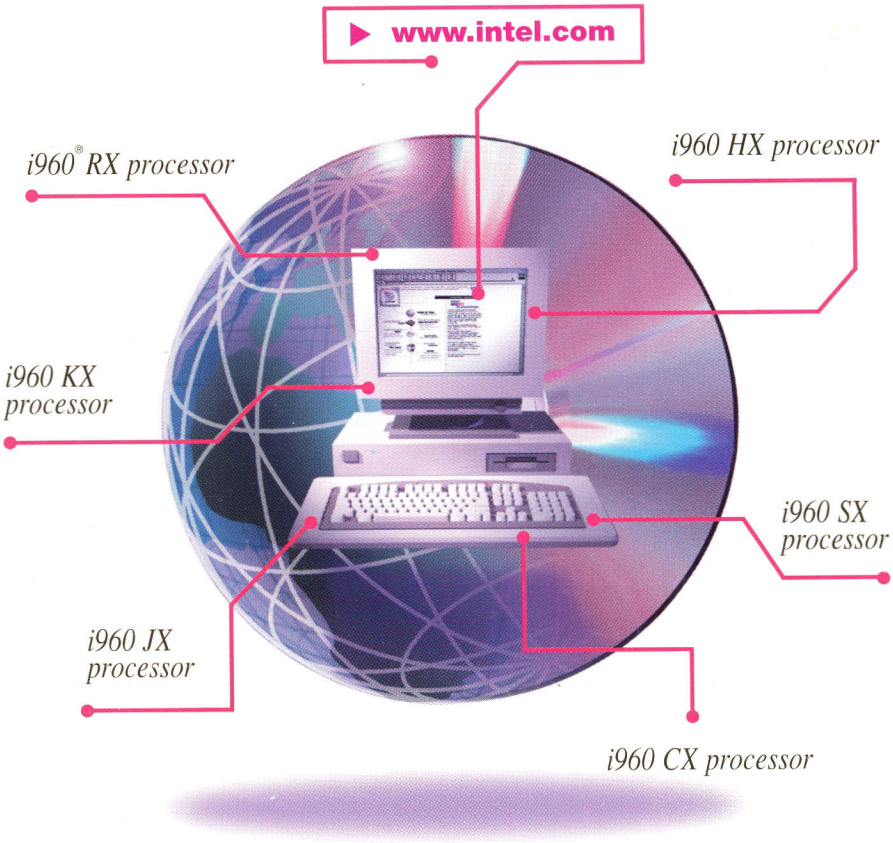
i960® Processor and Related Products



intel®

ARROW ELECTRONICS CANADA LTD.
SEMICONDUCTOR GROUP
171 SUPERIOR BOULEVARD
MISSISSAUGA, ONTARIO L5T 2L6
905/670/7769
FAX 905/670/7781

i960® Processor and Related Products

The i960® family of embedded processors are used in a wide range of embedded applications including imaging, networking, communications and intelligent I/O. Whatever your application there is an i960 processor to fit your cost and performance goals. This book contains detailed product specifications on the currently available processor versions. We've also included information on other related Intel products that can be used to enhance your design of speed your development.

Intel Order Number 272084-006
Printed in USA/0397/BC/HS
Semiconductor Products

Printed on Recycled Material

ISBN 1-55512-267-1

EAN 9 781555 122676 90000 >

intel®

LITERATURE

For additional information on Intel products in the U.S. or Canada, call Intel's Literature Center at (800) 548-4725 or write to:

INTEL LITERATURE
P.O. Box 7641
Mt. Prospect, IL 60056-7641

Literature and information about Intel's products is available on Intel's web site: **http://www.intel.com**

To order literature outside the U.S. and Canada contact your local sales office or visit Intel's web site.

CURRENT DATABOOKS

Product line databooks contain datasheets, application notes, article reprints and other design information. Databooks can be ordered in the U.S. and Canada by calling TAB/McGraw-Hill at (800) 262-4729; outside of the U.S. and Canada contact your local sales office.

New Titles for 1997

Title	Year	Intel Order Number	ISBN
Embedded Applications	1997	270648	1-55512-262-0
Embedded Microcontrollers	1997	270646	1-55512-263-9
Flash Memory (2 volume set)	1997	210830	1-55512-266-3
i960® Processors	1997	272084	1-55512-267-1
Intel's Quality System	1997	210997	1-55512-270-1
Networking	1997	297360	1-55512-264-7
Packaging	1997	240800	1-55512-269-8
Pentium® and Pentium Pro Processors and Related Products	1997	241732	1-55512-265-5
Platform Components (previously Peripherals)	1997	296467	1-55512-268-x

Titles Available from 1996

Title	Year	Intel Order Number	ISBN
Automotive	1996	231792	1-55512-257-4
Embedded Microprocessors	1996	272396	1-55512-249-3
Military	1996	210461	N/A

Intel Application Support Services

World Wide Web [URL: http://www.intel.com]

Intel's web site contains technical and product information that is available 24 hours a day! Also visit Intel's site for financials, history, current news and events, job opportunities, educational news and much, much more!

FTP Service [URL: ftp://download.intel.com/design]

Intel's FTP Service contains much of the technical and product documentation available on Intel's web site. These documents are available for you to download 24 hours a day.

Developer's Insight on CD-ROM

This quarterly publication contains a snapshot of the technical and product information contained in the Developer's Insight section of Intel's web site. This includes electronic copies of technical documents. To order your copy contact Intel's U.S. Literature Center or visit Intel's web site.

FaxBack

Technical and product information is available 24 hours a day via your fax machine!

To use FaxBack (for Intel components & systems), dial **(800) 525-3019** or 503-264-6835 (US/Canada/APAC/Japan) *or +44{0} 1793-496646 (Europe)* and follow the automated voice-prompt. Document orders will be faxed to the fax number you specify. For information on how the Intel Application Support team can help you, order our Customer Service Agreement, document #1201. Catalogs are updated as needed, so call for the latest information.

Bulletin Board System (BBS)

To use the Intel Application BBS (components & systems), dial **(503) 264-7999** (US/Canada/APAC/Japan) *or +44{0} 1793-432955 (Europe).* The BBS will support 1200-19200 baud rate modem. *Typical modem configuration: 14.4K baud rate, No Parity, 8 Data Bits, 1 Stop Bit.*

Dial 1-800-628-8686 or 916-356-7599 (U.S. & Canada) between 5 a.m. to 5 p.m. PST for help with Intel products. For customers not in the U.S. or Canada, please contact your local distributor.

Intel Literature Centers

Printed literature containing technical and product information may be obtained from Intel's Literature Centers.

U.S.	+1-800-548-4725	France	+44{0} 1793 421777
U.S. (from overseas)	+1-708-296-9333	Germany	+44{0} 1793 421333
England	+44{0} 1793 431 155	Japan (fax only)	+81{0} 120 47 88 32

Intel Distributors

Visit Intel's web site, check the back of an Intel databook or request one of the following distributor listing FaxBack documents: #4083 (U.S. Eastern Time Zone), #4084 (U.S. Central Time Zone), #4085 (Mountain Time Zone), #4086 (U.S. Alaska / Pacific Time Zone), #4209 (Europe) or #4403 (Canada).



i960® Processors and Related Products

i960® RX Processor, i960® HX Processor, i960® JX Processor, i960® CX Processor, i960® KX Processor, i960® SX Processor

1997



i960® Processors and Related Products

i960® RX Processor, i960® HX Processor,
i960® JX Processor, i960® CX Processor,
i960® KX Processor, i960® SX Processor

Information in this document is provided in connection with Intel products. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Except as provided in Intel's Terms and Conditions of Sale for such products, Intel assumes no liability whatsoever, and Intel disclaims any express or implied warranty, relating to sale and/or use of Intel products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right. Intel products are not intended for use in medical, life saving, or life sustaining applications.

Intel may make changes to specifications and product descriptions at any time, without notice.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Third-party brands and names are the property of their respective owners.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an ordering number and are referenced in this document, or other Intel literature, may be obtained from:

Intel Corporation
P.O. Box 7641
Mt. Prospect IL 60056-7641

or call 1-800-879-4683
or visit Intel's website at http:\\www.intel.com

Copyright © Intel Corporation 1996, 1997.

DATASHEET DESIGNATIONS

Intel uses various datasheet markings to designate each phase of the document as it relates to the product. The markings appear in the lower inside corner of each datasheet page. Following are the definitions of each marking:

Datasheet Marking	Description
Product Preview	Contains information on products in the design phase of development. Do not finalize a design with this information. Revised information will be published when the product becomes available.
Advanced Information	Contains information on products being sampled or in the initial production phase of development.*
Preliminary	Contains preliminary information on new products in production.*
No Marking	Contains information on products in full production.*

*Specifications within these datasheets are subject to change without notice. Verify with your local Intel sales office that you have the latest datasheet before finalizing a design.

November 1995

DATASHEET DESIGNATIONS

Intel uses various datasheet markings to designate each phase of the document as it relates to the product. The markings appear in the lower inside corner of each datasheet page. Following are the definitions of each marking:

Datasheet Marking	Description
Product Preview	Contains information on products in the design phase of development. Do not finalize a design with this information. Revised information will be published when the product becomes available.
Advanced Information	Contains information on products being sampled or in the initial production phase of development.*
Preliminary	Contains preliminary information on new products in production.*
No Marking	Contains information on products in full production.*

*Specifications within these datasheets are subject to change without notice. Verify with your local Intel sales office that you have the latest datasheet before finalizing a design.

November 1998

Microprocessor 1-347
80960KA Embedded 32-Bit Microprocessor 1-421
80960KB Embedded 32-Bit Microprocessor with Integrated Floating-Point Unit 1-423
80960SA Embedded 32-Bit Microprocessor with 16-Bit Burst Data Bus 1-425
80960SB Embedded 32-Bit Microprocessor with 16-Bit Burst Data Bus 1-427
AP-506 Designing for 80960Cx and 80960Hx Compatibility 1-429
AP-703 DRAM Controller for 33 MHz i960® CA/CF Microprocessors 1-459
AP-704 A Simple DRAM Controller for 25/16 MHz i960® CA/CF Microprocessors 1-499
AP-706 DRAM Controller for 40 MHz i960® CA/CF Microprocessors 1-539
AP-712 DRAM Controller for i960® JA/JF/JD Microprocessors 1-575
AP-716 80960Cx/80960Jx/80960Hx Architectural Comparison 1-607
AP-726 Interfacing the i960® Jx Microprocessor to the NEC µPD98401* Local ATM Segmentation and Reassembly (SAR) Chip 1-665

CHAPTER 2
Memories and Peripherals
28F016XS 16-Mbit (1Mbit x 16, 2 Mbit x 8) Synchronous Flash Memory 2-1
28F016XD 16-Mbit (1Mbit x 16) DRAM-Interface Flash Memory 2-7
28F016SV 16-Mbit (1Mbit x 16, 2 Mbit x 8) FlashFile™ Memory 2-11
AB-60 2/4/8-Mbit SmartVoltage Boot Block Flash Memory Family Overview 2-17
Interfacing the 28F016XS to the i960® Microprocessor Family 2-29
Value Series 100 Flash Memory Card 2-,4-,8-Megabyte 2-65
Series 2+ Flash Memory Cards 4-, 8-, 20- and 40-Megabyte 2-93
Series 2 Flash Memory Cards iMC002FLSA, iMC004FLSA, iMC010FLSA, iMC020FLSA 2-133

CHAPTER 3
Development Support Tools
i960® Microprocessor CTOOLS Application Development Tools 3-1
i960® Microprocessor Evaluation Platform 3-5

Alphanumeric Index

28F016SV 16-Mbit (1Mbit x 16, 2 Mbit x 8) FlashFile™ Memory 2-11
28F016XD 16-Mbit (1Mbit x 16) DRAM-Interface Flash Memory 2-7
28F016XS 16-Mbit (1Mbit x 16, 2 Mbit x 8) Synchronous Flash Memory 2-1
80960CA-33, -25, -16 32-Bit High-Performance Embedded Processor 1-345
80960CF-40, -33, -25, -16 32-Bit High-Performance Superscalar Embedded Microprocessor 1-347
80960HA/HD/HT 32-Bit High-Performance Superscalar Processor 1-69
80960JA/JF Embedded 32-Bit Microprocessor 1-167
80960JD Embedded 32-Bit Microprocessor 1-285
80960KA Embedded 32-Bit Microprocessor 1-421
80960KB Embedded 32-Bit Microprocessor with Integrated Floating-Point Unit 1-423
80960SA Embedded 32-Bit Microprocessor with 16-Bit Burst Data Bus 1-425
80960SB Embedded 32-Bit Microprocessor with 16-Bit Burst Data Bus 1-427
80L960JA/JF 3.3 V Embedded 32-Bit Microprocessor 1-229
AB-60 2/4/8-Mbit SmartVoltage Boot Block Flash Memory Family Overview 2-17
AP-506 Designing for 80960Cx and 80960Hx Compatibility 1-429
AP-703 DRAM Controller for 33 MHz i960® CA/CF Microprocessors 1-459
AP-704 A Simple DRAM Controller for 25/16 MHz i960® CA/CF Microprocessors 1-499
AP-706 DRAM Controller for 40 MHz i960® CA/CF Microprocessors 1-539
AP-712 DRAM Controller for i960® JA/JF/JD Microprocessors 1-575
AP-716 80960Cx/80960Jx/80960Hx Architectural Comparison 1-607
AP-726 Interfacing the i960® Jx Microprocessor to the NEC µPD98401* Local ATM Segmentation and Reassembly (SAR) Chip 1-665
i960® Microprocessor CTOOLS Application Development Tools 3-1
i960® Microprocessor Evaluation Platform 3-5
i960® RP I/O Processor 1-1
Interfacing the 28F016XS to the i960® Microprocessor Family 2-29
Series 2 Flash Memory Cards iMC002FLSA, iMC004FLSA, iMC010FLSA, iMC020FLSA 2-133
Series 2+ Flash Memory Cards 4-, 8-, 20- and 40-Megabyte 2-93
Value Series 100 Flash Memory Card 2-,4-,8-Megabyte 2-65

intel®

1

i960® Microprocessor Family

1

i960® Microprocessor Family

ADVANCE INFORMATION

i960® RP I/O PROCESSOR

• 33 MHz, 5.0 Volt Version (80960RP 33/5.0)
• Complies with PCI Local Bus Specification Revision 2.1

- **High Performance 80960JF Core**
  - **Sustained One Instruction/Clock Execution**
  - **4 Kbyte Two-Way Set-Associative Instruction Cache**
  - **2 Kbyte Direct-Mapped Data Cache**
  - **Sixteen 32-Bit Global Registers**
  - **Sixteen 32-Bit Local Registers**
  - **Programmable Bus Widths: 8-, 16-, 32-Bit**
  - **1 Kbyte Internal Data RAM**
  - **Local Register Cache (Eight Available Stack Frames)**
  - **Two 32-Bit On-Chip Timer Units**
- **PCI-to-PCI Bridge Unit**
  - **Primary and Secondary PCI Interfaces**
  - **Two 64-Byte Posting Buffers**
  - **Delayed and Posted Transaction Support**
  - **Forwards Memory, I/O, Configuration Commands from PCI Bus to PCI Bus**
- **Two Address Translation Units**
  - **Connects Local Bus to PCI Buses**
  - **Inbound/Outbound Address Translation Support**
  - **Direct Outbound Addressing Support**
- **Messaging Unit**
  - **Four Message Registers**
  - **Two Doorbell Registers**
  - **Four Circular Queues**
  - **1004 Index Registers**
- **Memory Controller**
  - **256 Mbytes of 32- or 36-Bit DRAM**
  - **Interleaved or Non-Interleaved DRAM**
  - **Fast Page-Mode DRAM Support**
  - **Extended Data Out and Burst**
  - **Extended Data Out DRAM Support**
  - **Two Independent Banks for SRAM / ROM / Flash (16 Mbytes/Bank; 8- or 32-Bit)**
- **DMA Controller**
  - **Three Independent Channels**
  - **PCI Memory Controller Interface**
  - **32-Bit Local Bus Addressing**
  - **64-Bit PCI Bus Addressing**
  - **Independent Interface to Primary and Secondary PCI Buses**
  - **132 Mbyte/sec Burst Transfers to PCI and Local Buses**
  - **Direct Addressing to and from PCI Buses**
  - **Unaligned Transfers Supported in Hardware**
  - **Two Channels Dedicated to Primary PCI Bus**
  - **One Channel Dedicated to Secondary PCI Bus**
- **I/O APIC Bus Interface Unit**
  - **Multiprocessor Interrupt Management for Intel Architecture CPUs (Pentium® and Pentium® Pro Processors)**
  - **Dynamic Interrupt Distribution**
  - **Multiple I/O Subsystem Support**
- **I^2C Bus Interface Unit**
  - **Serial Bus**
  - **Master/Slave Capabilities**
  - **System Management Functions**
- **Secondary PCI Arbitration Unit**
  - **Supports Six Secondary PCI Devices**
  - **Multi-priority Arbitration Algorithm**
  - **External Arbitration Support Mode**
- **Private PCI Device Support**
- **SuperBGA* Package**
  - **352 Ball-Grid Array (HL-PBGA)**

© INTEL CORPORATION, 1996
October 1996 Order Number: 272737-003

1960® RP I/O PROCESSOR

- 33 MHz, 5.0 Volt Version (80960RP 33/5.0)
- Complies with PCI Local Bus Specification Revision 2.1

- High Performance 80960JF Core
  - Sustained One Instruction/Clock Execution
  - 4 Kbyte Two-Way Set-Associative Instruction Cache
  - 2 Kbyte Direct-Mapped Data Cache
  - Sixteen 32-Bit Global Registers
  - Sixteen 32-Bit Local Registers
  - Programmable Bus Widths: 8-, 16-, 32-Bit
  - 1 Kbyte Internal Data RAM
  - Local Register Cache (Eight Available Stack Frames)
  - Two 32-Bit On-Chip Timer Units
- PCI-to-PCI Bridge Unit
  - Primary and Secondary PCI Interfaces
  - Two 64-Byte Posting Buffers
  - Delayed and Posted Transaction Support
  - Forwards Memory, I/O, Configuration Commands from PCI Bus to PCI Bus
- Two Address Translation Units
  - Connects Local Bus to PCI Buses
  - Inbound/Outbound Address Translation Support
  - Direct Outbound Addressing Support
- Messaging Unit
  - Four Message Registers
  - Two Doorbell Registers
  - Four Circular Queues
  - I₂O Index Registers
- Memory Controller
  - 256 Mbytes of 32- or 36-Bit DRAM
  - Interleaved or Non-Interleaved DRAM
  - Fast Page-Mode DRAM Support
  - Extended Data Out and Burst
  - Extended Data Out DRAM Support
  - Two Independent Banks for SRAM / ROM / Flash (16 Mbytes/Bank; 8- or 32-Bit)
- DMA Controller
  - Three Independent Channels
  - PCI Memory Controller Interface
  - 32-Bit Local Bus Addressing
  - 64-Bit PCI Bus Addressing
  - Independent Interface to Primary and Secondary PCI Buses
  - 132 Mbyte/sec Burst Transfers to PCI and Local Buses
  - Direct Addressing to and from PCI Buses
  - Unaligned Transfers Supported in Hardware
  - Two Channels Dedicated to Primary PCI Bus
  - One Channel Dedicated to Secondary PCI Bus
- I/O APIC Bus Interface Unit
  - Multiprocessor Interrupt Management for Intel Architecture CPUs (Pentium® and Pentium® Pro Processors)
  - Dynamic Interrupt Distribution
  - Multiple I/O Subsystem Support
- I²C Bus Interface Unit
  - Serial Bus
  - Master/Slave Capabilities
  - System Management Functions
- Secondary PCI Arbitration Unit
  - Supports Six Secondary PCI Devices
  - Multi-priority Arbitration Algorithm
  - External Arbitration Support Mode
- Private PCI Device Support
- SuperBGA* Package
  - 352 Ball-Grid Array (HL-PBGA)

© INTEL CORPORATION, 1998 October 1998 Order Number 272737-003

1.0 ABOUT THIS DOCUMENT

This data sheet contains advance information about Intel's i960 RP I/O processor at 5 Volts (referred to as *80960RP 33/5.0*), including a functional overview, mechanical data (package signal locations and simulated thermal characteristics), targeted electrical specifications (simulated), and bus functional waveforms. Detailed functional descriptions other than parametric performance is published in the *i960® RP Microprocessor User's Guide* (272736).

1.1 Solutions960® Program

Intel's *Solutions960®* program features a wide variety of development tools which support the i960 processor family. Many of these tools are developed by partner companies; some are developed by Intel, such as profile-driven optimizing compilers. For more information on these products, contact your local Intel representative.

1.2 Terminology

In this document, the following terms are used:

- *local bus* refers to the 80960RP 33/5.0's internal local bus, not the PCI local bus.
- *Primary and Secondary PCI buses* are the 80960RP 33/5.0's internal PCI buses which conform to PCI SIG specifications.
- *80960 core* refers to the 80960JF processor which is integrated into the 80960RP 33/5.0.

1.3 Additional Information Sources

Intel documentation is available from your local Intel Sales Representative or Intel Literature Sales.

Intel Corporation
Literature Sales
P.O. Box 7641
Mt. Prospect IL 60056-7641
1-800-879-4683

Table 1. Related Documentation

Document Title	Order / Contact
i960® RP Microprocessor User's Guide	Intel Order # 272736
i960® RP Processor: A Single-Chip Intelligent I/O Subsystem Technical Brief	Intel Order # 272738
i960® Jx Microprocessor User's Guide	Intel Order # 272483
80960RP Specification Update	Intel Order # 272918
PCI Local Bus Specification Revision 2.1	PCI Special Interest Group 1-800-433-5177
PCI-to-PCI Bridge Architecture Specification Revision 1.0	PCI Special Interest Group 1-800-433-5177
I²C Peripherals for Microcontrollers	Philips Semiconductor

1.4 Electronic Information

Intel's World-Wide Web (WWW) Location:	http://www.intel.com
Customer Support (US and Canada):	800-628-8686
FaxBack Service:	
US and Canada	800-628-2283
Europe	+44(0)793-496646
worldwide	916-356-3105
Application Bulletin Board Service:	
up to 14.4-Kbaud line, worldwide	916-356-3600
dedicated 2400-baud line, worldwide	916-356-7209
Europe	+44(0)793-496340

2.0 FUNCTIONAL OVERVIEW

As indicated in Figure 1, the 80960RP 33/5.0 combines many features with the 80960JF to create an intelligent I/O processor. Subsections following the figure briefly describe the main features; for detailed functional descriptions, refer to the *i960® RP Microprocessor User's Guide* (272736).

The PCI bus is an industry standard, high performance, low latency system bus that operates up to 132 Mbyte/s. The 80960RP 33/5.0, a multi-function PCI device, is fully compliant with the *PCI Local Bus Specification* Revision 2.1. Function 0 is the PCI-to-PCI bridge unit; Function 1 is the address translation unit.

The PCI-to-PCI bridge unit is the connection path between two independent 32-bit PCI buses and provides the ability to overcome PCI electrical load limits. The addition of the i960 core processor brings intelligence to the bridge.

The 80960RP 33/5.0, object code compatible with the i960 core processor, is capable of sustained execution at the rate of one instruction per clock.

The local bus, a 32-bit multiplexed burst bus, is a high-speed interface to system memory and I/O. A full complement of control signals simplifies the connection of the 80960RP 33/5.0 to external components. Physical and logical memory attributes are programmed via memory-mapped control registers (MMRs), an extension not found on the i960 Kx, Sx or Cx processors. Physical and logical configuration registers enable the processor to operate with all combinations of bus width and data object alignment.

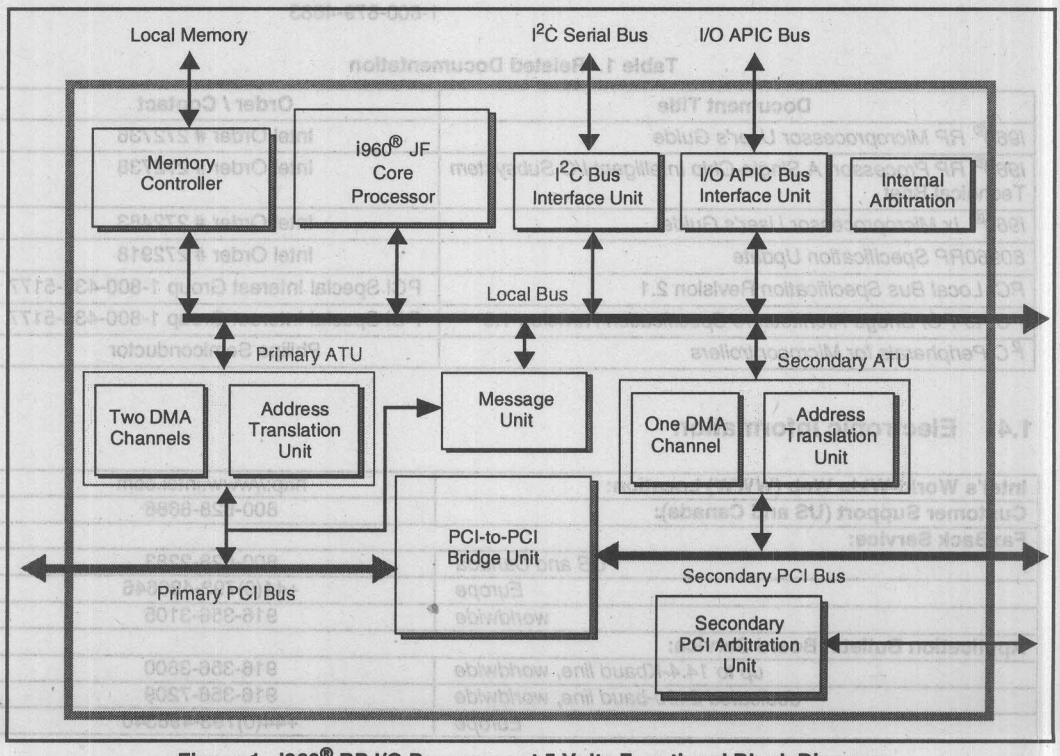


Figure 1. i960® RP I/O Processor at 5 Volts Functional Block Diagram

2.1 Key Functional Units

2.1.1 PCI-to-PCI Bridge Unit

The PCI-to-PCI bridge unit (referred to as "bridge") connects two independent PCI buses. It is fully compliant with the *PCI-to-PCI Bridge Architecture Specification* Revision 1.0 published by the PCI Special Interest Group. It allows certain bus transactions on one PCI bus to be forwarded to the other PCI bus. Dedicated data queues support high performance bandwidth on the PCI buses. The i960® RP I/O Processor at 5 Volts supports PCI 64-bit Dual Address Cycle (DAC) addressing.

The bridge has dedicated PCI configuration space that is accessible through the primary PCI bus.

2.1.2 Private PCI Device Support

A key design feature is that the 80960RP 33/5.0 explicitly supports private PCI devices on the secondary PCI bus without being detected by PCI configuration software. The bridge and Address Translation Unit work together to hide private devices from PCI configuration cycles and allow these devices to use a private PCI address space. The Address Translation Unit uses normal PCI configuration cycles to configure these devices.

2.1.3 DMA Controller

The DMA Controller supports low-latency, high-throughput data transfers between PCI bus agents and 80960 local memory. Three separate DMA channels accommodate data transfers: two for primary PCI bus, one for the secondary PCI bus. The DMA Controller supports chaining and unaligned data transfers. It is programmable only through the i960 core processor.

2.1.4 Address Translation Unit

The Address Translation Unit (ATU) allows PCI transactions direct access to the 80960RP 33/5.0 local memory. The 80960RP 33/5.0 has direct access to both PCI buses. The ATU supports transactions between PCI address space and 80960RP 33/5.0 address space.

Address translation is controlled through programmable registers accessible from both the PCI interface and the 80960 core. Dual access to registers allows flexibility in mapping the two address spaces.

2.1.5 Messaging Unit

The Messaging Unit (MU) provides data transfer between the PCI system and the 80960RP 33/5.0. It uses interrupts to notify each system when new data arrives. The MU has four messaging mechanisms. Each allows a host processor or external PCI device and the 80960RP 33/5.0 to communicate through message passing and interrupt generation. The four mechanisms are Message Registers, Doorbell Registers, Circular Queues, and Index Registers.

2.1.6 Memory Controller

The Memory Controller allows direct control of external memory systems, including DRAM, SRAM, ROM and Flash Memory. It provides a direct connect interface to memory that typically does not require external logic. It features programmable chip selects, a wait state generator and byte parity. External memory can be configured as PCI addressable memory or private processor memory.

2.1.7 I^2C Bus Interface Unit

The I^2C (Inter-Integrated Circuit) Bus Interface Unit allows the 80960 core to serve as a master and slave device residing on the I^2C bus. The I^2C bus is a serial bus developed by Philips Semiconductor consisting of a two pin interface. The bus allows the 80960RP 33/5.0 to interface to other I^2C peripherals and microcontrollers for system management functions. It requires a minimum of hardware for an economical system to relay status and reliability information on the I/O subsystem to an external device. For more information, see *I^2C Peripherals for Microcontrollers* (Philips Semiconductor)

2.1.8 I/O APIC Bus Interface Unit

The I/O APIC Bus Interface Unit provides an interface to the three-wire Advanced Programmable Interrupt Controller (APIC) bus that allows I/O APIC emulation in software. Interrupt messages can be sent on the bus and EOI messages can be received.

2.1.9 Secondary PCI Arbitration Unit

The Secondary PCI Arbitration Unit provides PCI arbitration for the secondary PCI bus. It includes a fairness algorithm with programmable priorities and six PCI Request and Grant signal pairs. This arbitration unit can also be disabled to allow for external arbitration.

2.2 i960 Core Features (80960JF)

The processing power of the 80960RP 33/5.0 comes from the 80960JF processor core. The 80960JF is a new, scalar implementation of the 80960 Core Architecture. Figure 2 shows a block diagram of the 80960JF Core processor.

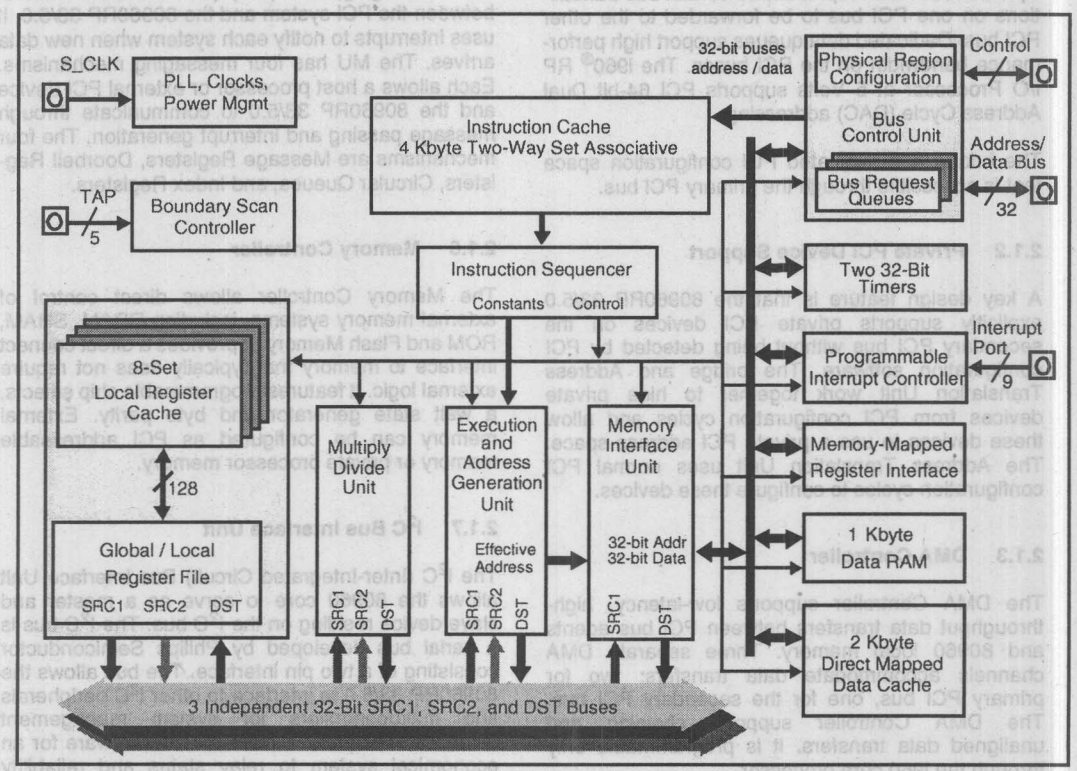


Figure 2. 80960JF Core Block Diagram

Factors that contribute to the 80960 family core's performance include:

- Single-clock execution of most instructions
- Independent Multiply/Divide Unit
- Efficient instruction pipeline minimizes pipeline break latency
- Register and resource scoreboarding allow overlapped instruction execution
- 128-bit register bus speeds local register caching
- 4 Kbyte two-way set-associative, integrated instruction cache
- 2 Kbyte direct-mapped, integrated data cache
- 1 Kbyte integrated data RAM delivers zero wait state program data

The 80960 core operates out of its own 32-bit address space, which is independent of the PCI address space. The local bus memory can be:

- Made visible to the PCI address space
- Kept private to the 80960 core
- Allocated as a combination of the two

2.2.1 Burst Bus

A 32-bit high-performance bus controller interfaces the 80960RP 33/5.0 to external memory and peripherals. The Bus Control Unit fetches instructions and transfers data on the local bus at the rate of up to four 32-bit words per six clock cycles. The external address/data bus is multiplexed.

Users may configure the 80960RP 33/5.0's bus controller to match an application's fundamental memory organization. Physical bus width is programmable for up to eight regions. Data caching is programmed through a group of logical memory templates and a defaults register. The Bus Control Unit's features include:

- Multiplexed external bus minimizes pin count
- 32-, 16- and 8-bit bus widths simplify I/O interfaces
- External ready control for address-to-data, data-to-data and data-to-next-address wait state types
- Little endian byte ordering
- Unaligned bus accesses performed transparently
- Three-deep load/store queue decouples the bus from the 80960 core

Upon reset, the 80960RP 33/5.0 conducts an internal self test. Before executing its first instruction, it performs an external bus confidence test by performing a checksum on the first words of the Initialization Boot Record.

2.2.2 Timer Unit

The timer unit (TU) contains two independent 32-bit timers that are capable of counting at several clock rates and generating interrupts. Each is programmed by use of the Timer Unit registers. These memory-mapped registers are addressable on 32-bit boundaries. The timers have a single-shot mode and auto-reload capabilities for continuous operation. Each timer has an independent interrupt request to the 80960RP 33/5.0's interrupt controller. The TU can generate a fault when unauthorized writes from user mode are detected.

2.2.3 Priority Interrupt Controller

Low interrupt latency is critical to many embedded applications. As part of its highly flexible interrupt mechanism, the 80960RP 33/5.0 exploits several techniques to minimize latency:

- Interrupt vectors and interrupt handler routines can be reserved on-chip
- Register frames for high-priority interrupt handlers can be cached on-chip
- The interrupt stack can be placed in cacheable memory space

2.2.4 Faults and Debugging

The 80960RP 33/5.0 employs a comprehensive fault model. The processor responds to faults by making implicit calls to a fault handling routine. Specific information collected for each fault allows the fault handler to diagnose exceptions and recover appropriately.

The processor also has built-in debug capabilities. Via software, the 80960RP 33/5.0 may be configured to detect as many as seven different trace event types. Alternatively, **mark** and **fmark** instructions can generate trace events explicitly in the instruction stream. Hardware breakpoint registers are also available to trap on execution and data addresses.

2.2.5 On-Chip Cache and Data RAM

Memory subsystems often impose substantial wait state penalties. The 80960RP 33/5.0 integrates considerable storage resources on-chip to decouple CPU execution from the external bus. It also includes a 4 Kbyte instruction cache, a 2 Kbyte data cache and 1 Kbyte data RAM.

2.2.6 Local Register Cache

The 80960RP 33/5.0 rapidly allocates and deallocates local register sets during context switches. The processor needs to flush a register set to the stack only when it saves more than seven sets to its local register cache.

2.2.7 Test Features

The 80960RP 33/5.0 incorporates numerous features that enhance the user's ability to test both the processor and the system to which it is attached. These features include ONCE (On-Circuit Emulation) mode and Boundary Scan (JTAG).

The 80960RP 33/5.0 provides testability features compatible with IEEE Standard Test Access Port and Boundary Scan Architecture (IEEE Std. 1149.1).

One of the boundary scan instructions, HIGHZ, forces the processor to float all its output pins (ONCE mode). ONCE mode can also be initiated at reset without using the boundary scan mechanism.

ONCE mode is useful for board-level testing. This feature allows a mounted 80960RP 33/5.0 to electrically "remove" itself from a circuit board. This mode allows system-level testing where a remote tester can exercise the processor system.

The test logic does not interfere with component or system behavior and ensures that components function correctly, and also the connections between various components are correct.

The JTAG Boundary Scan feature is an alternative to conventional "bed-of-nails" testing. It can examine connections that might otherwise be inaccessible to a test system.

2.2.8 Memory-Mapped Control Registers

The 80960RP 33/5.0 is compliant with 80960 family architecture and has the added advantage of memory-mapped, internal control registers not found on the 80960Kx, Sx or Cx processors. This feature provides software an interface to easily read and modify internal control registers.

Each memory-mapped, 32-bit register is accessed via regular memory-format instructions. The processor ensures that these accesses do not generate external bus cycles.

2.2.9 Instructions, Data Types and Memory Addressing Modes

As with all 80960 family processors, the 80960RP 33/5.0 instruction set supports several different data types and formats:

- Bit
- Bit fields
- Integer (8-, 16-, 32-, 64-bit)
- Ordinal (8-, 16-, 32-, 64-bit unsigned integers)
- Triple word (96 bits)
- Quad word (128 bits)

The 80960RP 33/5.0 provides a full set of addressing modes for C and assembly:

- Two Absolute modes
- Five Register Indirect modes
- Index with displacement mode
- IP with displacement mode

Table 2 shows the available instructions.

Table 2. 80960RP 33/5.0 Instruction Set

Data Movement	Arithmetic	Logical	Bit, Bit Field and Byte
Load	Add	And	Set Bit
Store	Subtract	Not And	Clear Bit
Move	Multiply	And Not	Not Bit
Conditional Select	Divide	Or	Alter Bit
Load Address	Remainder	Exclusive Or	Scan For Bit
	Modulo	Not Or	Span Over Bit
	Shift	Or Not	Extract
	Extended Shift	Nor	Modify
	Extended Multiply	Exclusive Nor	Scan Byte for Equal
	Extended Divide	Not	Byte Swap
	Add with Carry	Nand	
	Subtract with Carry		
	Conditional Add		
	Conditional Subtract		
	Rotate		
Comparison	**Branch**	**Call/Return**	**Fault**
Compare	Unconditional Branch	Call	Conditional Fault
Conditional Compare	Conditional Branch	Call Extended	Synchronize Faults
Compare and Increment	Compare and Branch	Call System	
Compare and Decrement		Return	
Test Condition Code		Branch and Link	
Check Bit			
Debug	**Processor Management**	**Atomic**	
Modify Trace Controls	Flush Local Registers	Atomic Add	
Mark	Modify Arithmetic Controls	Atomic Modify	
Force Mark	Modify Process Controls		
	Halt		
	System Control		
	Cache Control		
	Interrupt Control		

3.0 PACKAGE INFORMATION

3.1 Package Introduction

The i960® RP I/O Processor at 5 Volts is offered in a SuperBGA* Ball Grid Array (HL-PBGA) package. This is a perimeter array package with four rows of ball connections in the outer area of the package. See Figure 4, 352L HL-PBGA Package Diagram (Bottom View) (pg. 22).

Section 3.1.1, Functional Signal Definitions describes signal function; Section 3.1.2, 352-Lead HL-PBGA Package (pg. 21) defines the signal and ball locations.

3.1.1 Functional Signal Definitions

Table 3 presents the legend for interpreting the Type Field in the following tables. Table 4 defines signals associated with the bus interface. Table 5 defines signals associated with basic control and test functions. Table 6 defines signals associated with the Interrupt Unit. Table 7 defines PCI signals. Table 8 defines Memory Controller signals. Table 9 defines DMA, APIC and I^2C signals. Table 10 defines clock signals. Table 11 defines ICE signals.

Table 3. Signal Type Definition

Symbol	Description
I	Input signal only.
O	Output signal only.
I/O	Signal can be either an input or output.
OD	Open Drain signal.
–	Signal must be connected as described.
S (...)	Synchronous. Inputs must meet setup and hold times relative to S_CLK. S(E) Edge sensitive input S(L) Level sensitive input
A (...)	Asynchronous. Inputs may be asynchronous relative to S_CLK. A(E) Edge sensitive input A(L) Level sensitive input
R (...)	While the P_RST# signal is asserted, the signal: R(1) is driven to V_{CC} R(0) is driven to V_{SS} R(Q) is a valid output R(Z) Floats R(H) is pulled up to V_{CC} R(X) is driven to an unknown state
H (...)	While the 80960RP 33/5.0 is in the hold state, the signal: H(1) is driven to V_{CC} H(0) is driven to V_{SS} H(Q) Maintains previous state or continues to be a valid output H(Z) Floats
P (...)	While the 80960RP 33/5.0 is halted, the signal: P(1) is driven to V_{CC} P(0) is driven to V_{SS} P(Q) Maintains previous state or continues to be a valid output
K (...)	While the Secondary PCI Bus is being parked, the signal: K(Z) Floats K(Q) Maintains previous state or continues to be a valid output

Table 4. Signal Descriptions (Sheet 1 of 5)

NAME	TYPE	DESCRIPTION
AD31:0	I/O S(L) R(Z) H(Z) P(Q)	**ADDRESS / DATA BUS** carries 32-bit physical addresses and 8-, 16- or 32-bit data to and from memory. During an address (T_a) cycle, bits 2-31 contain a physical word address (bits 0-1 indicate SIZE; see below). During a data (T_d) cycle, read or write data is present on one or more contiguous bytes, comprising AD31:24, AD23:16, AD15:8 and AD7:0. During write operations, unused signals are driven to determinate values. **SIZE**, which comprises bits 0-1 of the AD lines during a T_a cycle, specifies the number of data transfers during the bus transaction on the local bus. When the DMA or ATUs initiate data transfers, transfer size shown below is **not** valid. **AD1 AD0 Bus Transfers** 0 0 1 Transfer 0 1 2 Transfers 1 0 3 Transfers 1 1 4 Transfers When the 80960RP 33/5.0 enters Halt mode and the previous bus operation was: • write — AD31:2 are driven with the last data value on the AD bus. • read — AD31:2 are driven with the last address value on the AD bus. Typically, AD1:0 reflect the SIZE information of the last bus transaction (either instruction fetch or load/store) that was executed before entering Halt mode.
ADS#	O R(1) H(Z) P(1)	**ADDRESS STROBE** indicates a valid address and the start of a new bus access. The processor asserts ADS# for the entire T_a cycle. External bus control logic typically samples ADS# at the end of the cycle.
ALE	O R(0) H(Z) P(0)	**ADDRESS LATCH ENABLE** indicates the transfer of a physical address. ALE is asserted during a T_a cycle and deasserted before the beginning of the T_d state. It is active HIGH and floats to a high impedance state during a hold cycle (T_h).
BLAST#	O R(1) H(Z) P(1)	**BURST LAST** indicates the last transfer in a bus access. BLAST# is asserted in the last data transfer of burst and non-burst accesses. BLAST# remains active while wait states are detected via the LRDYRCV# or RDYRCV# signal on the memory controller. BLAST# becomes inactive after the final data transfer in a bus cycle. BLAST# has a weak internal pullup which is active during reset to ensure normal operation when the signal is not connected. 0 = Last Data Transfer 1 = Not the Last Data Transfer

Table 4. Signal Descriptions (Sheet 2 of 5)

NAME	TYPE	DESCRIPTION
BE3:0#	O R(1) H(Z) P(1)	**BYTE ENABLES** select which of up to four data bytes on the bus participate in the current bus access. Byte enable encoding depends on the bus width of the memory region accessed: *32-bit bus:* BE3# enables data on AD31:24 BE2# enables data on AD23:16 BE1# enables data on AD15:8 BE0# enables data on AD7:0 *16-bit bus:* BE3# becomes Byte High Enable (enables data on AD15:8) BE2# is not used (state is high) BE1# becomes Address Bit 1 (A1) (increments with the assertion of LRDY# or RDYRCV#) BE0# becomes Byte Low Enable (enables data on AD7:0) *8-bit bus:* BE3# is not used (state is high) BE2# is not used (state is high) BE1# becomes Address Bit 1 (A1) (increments with the assertion of LRDY# or RDYRCV#) BE0# becomes Address Bit 0 (A0) (increments with the assertion of LRDY# or RDYRCV#) The processor asserts byte enables, byte high enable and byte low enable during T_a. Since unaligned bus requests are split into separate bus transactions, these signals do not toggle during a burst (32-bit bus only) from the i960 core processor; they do toggle for DMA and ATU cycles. They remain active through the last T_d cycle.
DEN#	O R(1) H(Z) P(1)	**DATA ENABLE** indicates data transfer cycles during a bus access. DEN# is asserted at the start of the first data cycle in a bus access and deasserted at the end of the last data cycle. DEN# is used with DT/R# to provide control for data transceivers connected to the data bus. DEN# has a weak internal pullup which is active during reset to ensure normal operation when the signal is not connected. 0 = Data Cycle 1 = Not a Data Cycle

Table 4. Signal Descriptions (Sheet 3 of 5)

NAME	TYPE	DESCRIPTION
D/C#/ RST_MODE#	I/O R(H) H(Z) P(Q)	**DATA/CODE/RESET_MODE** indicates that a bus access is a data access or an instruction access. D/C# has the same timing as W/R#. 0 = Instruction Access 1 = Data Access The RST_MODE# signal is sampled at Primary PCI bus reset to determine whether the 80960 core is to be held in reset. When RST_MODE# is high, the 80960RP 33/5.0 begins initialization immediately following the deassertion of P_RST. When RST_MODE is low, the 80960 core remains in reset until the 80960 core reset bit is cleared in the extended bridge control register. This signal has a weak internal pullup that is active during reset to ensure normal operation when the signal is left unconnected. 0 = RST_MODE enabled 1 = RST_MODE not enabled While the 80960 core is in reset, all peripherals may be accessed from the primary or secondary PCI buses depending on the status of the WIDTH/HLTD1/RETRY/ signal.
DT/R#	O R(0) H(Z) P(Q)	**DATA TRANSMIT / RECEIVE** indicates the direction of data transfer to and from the address/data bus. It is low during T_a and T_w/T_d cycles for a read; it is high during T_a and T_w/T_d cycles for a write. DT/R# never changes state when DEN# is asserted. 0 = Receive 1 = Transmit
LOCK#/ONCE#	I/O S(L) R(H) H(Z) P(Q)	**BUS LOCK** indicates that an atomic read-modify-write operation is in progress. The LOCK# output is asserted in the first clock of an atomic operation and deasserted in the last data transfer of the sequence. The processor does not grant HOLDA while asserting LOCK#. This prevents external agents from accessing memory involved in semaphore operations. 0 = Atomic Read-Modify-Write in Progress 1 = No Atomic Read-Modify-Write in Progress **ONCE MODE:** The processor samples the ONCE input during reset. When ONCE# is asserted LOW at the end of reset, the processor enters ONCE mode, stops all clocks and floats all output signals. LOCK#/ONCE# has a weak internal pullup which is active during reset to ensure normal operation when the signal is not connected. 0 = ONCE Mode Enabled 1 = ONCE Mode Not Enabled
LRDYRCV#	O R(1) H(Q) P(Q)	**LOCAL READY/RECOVER**, generated by the 80960RP 33/5.0's memory controller unit, is an output version of the READY/RECOVER (RDYRCV#) signal. Refer to the RDYRCV# signal description.

Table 4. Signal Descriptions (Sheet 4 of 5)

NAME	TYPE	DESCRIPTION
HOLD	I S(L)	**HOLD** is a request from an external bus master to acquire the bus. When the processor receives HOLD and grants bus control to another master, it asserts HOLDA, floats the address/data and control lines and enters the T_h state. When HOLD is deasserted, the processor deasserts HOLDA and enters either the T_i or T_a state, resuming control of the address/data and control lines. See Figure 31, HOLD/HOLDA Waveform For Bus Arbitration (pg. 61). 0 = No Hold Request 1 = Hold Requested
HOLDA	O R(0) H(1) P(Q)	**HOLD ACKNOWLEDGE** indicates to an external bus master that the processor has relinquished bus control. The processor can grant HOLD requests and enter the T_h state and while halted as well as during regular operation. See Figure 31, HOLD/HOLDA Waveform For Bus Arbitration (pg. 61). 0 = No Hold Acknowledged 1 = Hold Acknowledged
RDYRCV#	I S(L)	**READY/RECOVER** is only used in systems that use an external memory controller (and do not use the 80960RP 33/5.0's memory controller unit). This signal indicates that data on AD lines can be sampled or removed. When RDYRCV# is not asserted during a T_d cycle, the T_d cycle extends to the next cycle by inserting a wait state (T_w). 0 = Sample Data 1 = Do Not Sample Data RDYRCV# has an alternate function during the recovery (T_r) state. The processor continues to insert recovery states until it samples the signal HIGH. This gives slow external devices more time to float their buffers before the processor drives addresses. 0 = Insert Wait States 1 = Recovery Complete When using the internal memory controller, connect this signal to V_{CC} through a 2.7 Kohm resistor.
W/R#	O R(0) H(Z) P(Q)	**WRITE/READ** specifies during a T_a cycle whether the operation is a write or read. It is latched on-chip and remains valid during T_d cycles. 0 = Read 1 = Write
WIDTH/ HLTD0	I/O R(H) H(Z) P(Q)	**WIDTH** denotes the physical memory attributes for a bus transaction in conjunction with WIDTH/HLTD1/RETRY: **WIDTH/HLTD1/RETRY** — **WIDTH/HLTD0** 0 — 0 — 8 Bits Wide 0 — 1 — 16 Bits Wide 1 — 0 — 32 Bits Wide 1 — 1 — Undefined WIDTH/HLTD0 has a weak internal pullup which is active during reset to ensure normal operation when the signal is not connected. HLTD0 signal name has no function in the 80960RP 33/5.0; the signal name is included for 80960JF naming convention compatibility.

Table 4. Signal Descriptions (Sheet 5 of 5)

NAME	TYPE	DESCRIPTION
WIDTH/ HLTD1/ RETRY	I/O R(H) H(Z) P(Q)	**WIDTH** denotes the physical memory attributes for a bus transaction in conjunction with the WIDTH/HLTD0 signal. Refer to description above. RETRY is sampled at Primary PCI bus reset to determine when the Primary PCI interface is disabled. When high, the Primary PCI interface disables PCI configuration cycles by signaling a RETRY until the Extended Bridge Control Register's Configuration Cycle Disable bit is cleared. When low, the Primary PCI interface allows configuration cycles to occur. WIDTH/HLTD1/RETRY has a weak internal pullup which is active during reset to ensure normal operation when the signal is not connected. HLTD1 signal name has no function in the 80960RP 33/5.0; the signal name is included for 80960JF naming convention compatibility.

Table 5. Power Requirement, Processor Control and Test Signal Descriptions (Sheet 1 of 2)

NAME	TYPE	DESCRIPTION
FAIL#	O R(0) H(Q) P(1)	**FAIL** indicates a failure of the processor's built-in self-test performed during initialization. FAIL# Is asserted immediately upon reset and toggles during self-test to indicate the status of individual tests: • When self-test passes, the processor deasserts FAIL# and commences operation from user code. • When self-test fails, the processor asserts FAIL# and then stops executing. Self-test failing does not cause the bridge to stop execution. 0 = Self Test Failed 1 = Self Test Passed
L_RST#	O	**LOCAL BUS RESET** notifies external devices that the local bus has reset.
STEST	I S(L)	**SELF TEST** enables or disables the processor's internal self-test feature at initialization. STEST is examined at the end of P_RST#. When STEST is asserted, the processor performs its internal self-test and the external bus confidence test. When STEST is deasserted, the processor performs only the external bus confidence test. 0 = Self Test Disabled 1 = Self Test Enabled
TCK	I	**TEST CLOCK** is a CPU input that provides the clocking function for IEEE 1149.1 Boundary Scan Testing (JTAG). State information and data are clocked into the processor on the rising edge; data is clocked out of the processor on the falling edge.
TDI	I S(L)	**TEST DATA INPUT** is the serial input signal for JTAG. TDI is sampled on the rising edge of TCK, during the SHIFT-IR and SHIFT-DR states of the Test Access Port. This signal has a weak internal pullup which is active during reset to ensure normal operation when the signal is not connected.
TDO	O R(Q) H(Q) P(Q)	**TEST DATA OUTPUT** is the serial output signal for JTAG. TDO is driven on the falling edge of TCK during the SHIFT-IR and SHIFT-DR states of the Test Access Port. At other times, TDO floats.

Table 5. Power Requirement, Processor Control and Test Signal Descriptions (Sheet 2 of 2)

NAME	TYPE	DESCRIPTION
TMS	I S(L)	**TEST MODE SELECT** is sampled at the rising edge of TCK to select the operation of the test logic for IEEE 1149.1 Boundary Scan testing. This signal has a weak internal pullup which is active during reset to ensure normal operation when the signal is not connected.
TRST#	I A(L)	**TEST RESET** asynchronously resets the Test Access Port (TAP) controller function of IEEE 1149.1 Boundary Scan testing (JTAG). When using the Boundary Scan feature, connect a pulldown resistor (1.5K ohm) between this signal and V_{SS}. When TAP is not used, this signal must be connected to V_{SS}; however, no resistor is required. The signal has a weak internal pullup which must be overcome during reset to ensure normal operation.
V_{CC}	–	**POWER.** Connect to a 5.0 Volt V_{CC} board plane.
V_{SS}	–	**GROUND.** Connect to a V_{SS} board plane.
N.C.	–	**NO CONNECT.** Do not make electrical connections to these balls.
$V_{CCPLL}3:1$	I	**PLL POWER.** For external connection to a 5 V V_{CC} board plane. Power to PLLs may require external filtering.

Table 6. Interrupt Unit Signal Descriptions

NAME	TYPE	DESCRIPTION
NMI#	I A(L)	**NON-MASKABLE INTERRUPT** causes a non-maskable interrupt event to occur. NMI# is the highest priority interrupt source and is level-detect. When NMI# is unused, it is recommended that you connect it to V_{CC}.
S_INT[A:D]#/ XINT3:0#	I A(L)	**SECONDARY PCI BUS INTERRUPT**[1] requests an interrupt. S_INTx# assertion and deassertion is asynchronous to S_CLK. A device asserts S_INTx# when requesting attention from its device driver. When S_INTx# is asserted, it remains asserted until the device driver clears the pending request. S_INTx# Interrupts are level sensitive. **EXTERNAL INTERRUPT.** External devices use this signal to request an interrupt service. These signals operate in dedicated mode, where each signal is assigned a dedicated interrupt level. The S_INT[A:D]#/XINT3:0# signals can be directed as follows: **Sec. PCI** **Primary PCI** **80960 Core Processor** S_INTA# ⇒ P_INTA# or XINT0# S_INTB# ⇒ P_INTB# or XINT1# S_INTC# ⇒ P_INTC# or XINT2# S_INTD# ⇒ P_INTD# or XINT3#
XINT7:4#	I A(L)	**EXTERNAL INTERRUPT.** External devices use this signal to request an interrupt service. These signals operate in dedicated mode, where each signal is assigned a dedicated interrupt level.

NOTE:

1. PCI signal functions are summarized in this data sheet; refer to the *PCI Local Bus Specification Revision 2.1* for a more complete definition.

Table 7. PCI Signal Descriptions (Sheet 1 of 3)

NAME	TYPE	DESCRIPTION[1]
P_AD31:0	I/O K(Q) R(Z)	**PRIMARY PCI ADDRESS/DATA** is the primary multiplexed PCI address and data bus.
P_C/BE3:0#	I/O K(Q) R(Z)	**PRIMARY PCI BUS COMMAND** and **BYTE ENABLE** signals are multiplexed on the same PCI signals. During an address phase, P_C/BE3:0# define the bus command. During a data phase, P_C/BE3:0# are used as byte enables.
P_DEVSEL#	I/O K(Z) R(Z)	**PRIMARY PCI BUS DEVICE SELECT** is driven by a target agent that has successfully decoded the address. As an input, it indicates whether or not an agent has been selected.
P_FRAME#	I/O K(Z) R(Z)	**PRIMARY PCI BUS CYCLE FRAME** is asserted to indicate the beginning and duration of an access on the Primary PCI bus.
P_GNT#	I R(Z)	**PRIMARY PCI BUS GRANT** indicates to the agent that access to the bus has been granted. This is a point-to-point signal.
P_IDSEL	I S(L)	**PRIMARY PCI BUS INITIALIZATION DEVICE SELECT** selects the 80960RP 33/5.0 during a Configuration Read or Write command on the primary PCI bus.
P_INT[A:D]#	O OD R(Z)	**PRIMARY PCI BUS INTERRUPT** requests an interrupt. The assertion and deassertion of P_INTx# is asynchronous to S_CLK. A device asserts its P_INTx# line when requesting attention from its device driver. Once the P_INTx# signal is asserted, it remains asserted until the device driver clears the pending request. P_INTx# Interrupts are level sensitive.
P_IRDY#	I/O K(Z) R(Z)	**PRIMARY PCI BUS INITIATOR READY** indicates the initiating agent's (bus master's) ability to complete the current data phase of the transaction.
P_LOCK#	I S(L)	**PRIMARY PCI BUS LOCK** indicates an atomic operation that may require multiple transactions to complete.
P_PAR	I/O K(Q) R(Z)	**PRIMARY PCI BUS PARITY**. This signal ensures even parity across P_AD31:0 and P_C/BE3:0. All PCI devices must provide a parity signal.
P_PERR#	I/O K(Z) R(Z)	**PRIMARY PCI BUS PARITY ERROR** is used for reporting data parity errors during all PCI transactions except a special cycle.
P_REQ#	O K(Q) R(Z)	**PRIMARY PCI BUS REQUEST** indicates to the arbiter that this agent desires use of the bus. This is a point to point signal.

NOTE:

1. PCI signal functions are summarized in this data sheet; refer to the *PCI Local Bus Specification Revision 2.1* for a more complete definition.

Table 7. PCI Signal Descriptions (Sheet 2 of 3)

NAME	TYPE	DESCRIPTION[1]
P_RST#	I A(L)	**PRIMARY RESET** brings 80960RP 33/5.0 to a consistent state. When P_RST# is asserted: • PCI output signals are driven to a known consistent state. • PCI bus interface output signals are three-stated. • open drain signals such as P_SERR# are floated. • S_RST# asserts. P_RST# may be asynchronous to S_CLK when asserted or deasserted. Although asynchronous, deassertion must be guaranteed to be a clean, bounce-free edge.
P_SERR#	I/O OD K(Z) R(Z)	**PRIMARY PCI BUS SYSTEM ERROR** reports address and data parity errors on the special cycle command, or any other system error where the result would be catastrophic.
P_STOP#	I/O K(Z) R(Z)	**PRIMARY PCI BUS STOP** indicates that the current target is requesting the master to stop the current transaction on the primary PCI bus.
P_TRDY#	I/O K(Z) R(Z)	**PRIMARY PCI BUS TARGET READY** indicates the target agent's (selected device's) ability to complete the current data phase of the transaction.
S_AD31:0	I/O R(0)	**SECONDARY PCI ADDRESS/DATA** is the secondary multiplexed PCI address and data bus. A bus transaction consists of an address phase followed by one or more data phases.
S_C/BE3:0#	I/O R(0)	**SECONDARY PCI BUS COMMAND and BYTE ENABLE** signals are multiplexed on the same PCI signals. During an address phase, S_C/BE3:0# define the bus command. During a data phase, S_C/BE3:0# are used as byte enables.
S_DEVSEL#	I/O R(Z)	**SECONDARY PCI BUS DEVICE SELECT** is driven by a target agent that has successfully decoded the address. As an input, it indicates whether or not an agent has been selected.
S_FRAME#	I/O R(Z)	**SECONDARY PCI BUS CYCLE FRAME** is asserted to indicate the beginning and duration of an access on the Secondary PCI bus.
S_GNT0#/ S_REQ#	O R(Z)	**SECONDARY PCI BUS GRANT0** is a grant signal sent to device 0 on the secondary PCI bus when the internal Secondary PCI Bus Arbiter is enabled. **SECONDARY PCI BUS REQUEST** is the request signal for the 80960RP 33/5.0 when the arbiter is disabled.
S_GNT5:1#	O R(Q)	**SECONDARY PCI BUS GRANT** are grant signals sent to devices 1-5 on the secondary PCI bus.
S_IDSEL	I S(L)	**SECONDARY PCI BUS INITIALIZATION DEVICE SELECT** selects the 80960RP 33/5.0 during a Configuration Read or Write command on the secondary PCI bus.

NOTE:

1. PCI signal functions are summarized in this data sheet; refer to the *PCI Local Bus Specification Revision 2.1* for a more complete definition.

Table 7. PCI Signal Descriptions (Sheet 3 of 3)

NAME	TYPE	DESCRIPTION[1]
S_IRDY#	I/O R(Z)	**SECONDARY PCI BUS INITIATOR READY** indicates the initiating agent's (bus master's) ability to complete the current data phase of the transaction.
S_LOCK#	I/O R(Z)	**SECONDARY PCI BUS LOCK** indicates the need to perform an atomic operation on the secondary PCI bus.
S_PAR	I/O R(0)	**SECONDARY PCI BUS PARITY.** This signal ensures even parity across S_AD31:0 and S_C/BE3:0. All PCI devices must provide a parity signal.
S_PERR#	I/O R(Z)	**SECONDARY PCI BUS PARITY ERROR** is used for reporting data parity errors during all PCI transactions except a special cycle.
S_REQ0#/ S_GNT#	I	**SECONDARY PCI BUS REQUEST0** is a request signal from device 0 on the secondary PCI bus when the internal Secondary PCI Bus Arbiter is enabled. **SECONDARY PCI BUS GRANT** is the grant signal for the 80960RP 33/5.0 when the arbiter is disabled.
S_RST#	O R(Q)	**SECONDARY PCI BUS RESET** is an output based on P_RST#. It brings PCI-specific registers, sequencers, and signals to a consistent state. When P_RST# is asserted, it causes S_RST# to assert, and: • PCI output signals are driven to a known consistent state. • PCI bus interface output signals are three-stated. • open drain signals such as S_SERR# are floated. S_RST# may be asynchronous to S_CLK when asserted or deasserted.
S_SERR#	I/O OD R(Z)	**SECONDARY PCI BUS SYSTEM ERROR** reports address and data parity errors on the special cycle command, or any other system error where the result would be catastrophic.
S_STOP#	I/O R(Z)	**SECONDARY PCI BUS STOP** indicates that the current target is requesting the master to stop the current transaction on the secondary PCI bus.
S_TRDY#	I/O R(Z)	**SECONDARY PCI BUS TARGET READY** indicates the target agent's (selected device's) ability to complete the current data phase of the transaction.
S_REQ4:1#	I S(L)	**SECONDARY PCI BUS REQUEST 4:1** are request signals from devices 1-4 on the secondary PCI bus.
S_REQ5#/ S_ARB_EN	I S(L)	**SECONDARY PCI BUS REQUEST 5** is the request signal from device 5 on the secondary PCI bus. **SECONDARY PCI BUS ARBITER ENABLE** defines the power-up status of the internal secondary arbitration unit. A valid high at the deassertion of P_RST# enables the internal secondary arbiter. A valid low at the deassertion of P_RST# disables the internal secondary arbiter.

NOTE:

1. PCI signal functions are summarized in this data sheet; refer to the *PCI Local Bus Specification Revision 2.1* for a more complete definition.

Table 8. Memory Controller Signal Descriptions (Sheet 1 of 2)

NAME	TYPE	DESCRIPTION
CAS7:0#	O R(1) H(Q) P(Q)	**COLUMN ADDRESS STROBE** signals are used for DRAM accesses and are asserted when the MA11:0 signals contain a valid column address. CAS7:0# signals are asserted during refresh. *Non-Interleaved Operation:* CAS0# = BE0# lane access CAS1# = BE1# lane access CAS2# = BE2# lane access CAS3# = BE3# lane access *Interleaved Operation:* CAS0# = BE0# Even leaf lane access CAS1# = BE1# Even leaf lane access CAS2# = BE2# Even leaf lane access CAS3# = BE3# Even leaf lane access CAS4# = BE0# Odd leaf lane access CAS5# = BE1# Odd leaf lane access CAS6# = BE2# Odd leaf lane access CAS7# = BE3# Odd leaf lane access
CE1:0#	O R(1) H(Q) P(Q)	**CHIP ENABLE** signals indicate an access to one of the two SRAM/ FLASH/ ROM memory banks. CE0# and CE1# are never asserted at the same time. These signals are valid during the entire memory operation. CE0# is asserted for accesses to memory bank 0. CE1# is asserted for accesses to memory bank 1.
DALE1:0	O R(0) H(Q) P(Q)	**DRAM ADDRESS LATCH ENABLE** signals support external address demultiplexing of the MA11:0 address lines for interleaved DRAM systems. Use these to directly interface to '373' type latches. These signals are only valid for accesses to interleaved memory systems. DALE0 is asserted during a valid even leaf address. DALE1 is asserted during a valid odd leaf address.
DP3:0	I/O R(X) H(Q) P(Q)	**DATA PARITY** carries the parity information for DRAM accesses. Each parity bit corresponds to a group of 8 data bus signals as follows: DP0 — AD7:0 DP2 — AD23:16 DP1 — AD15:8 DP3 — AD31:24 The memory controller generates parity information for local bus writes during data cycles. During read data cycles, the memory controller checks parity and provides notification of parity errors on the clock following the data cycle. Parity checking and polarity are user-programmable. Parity generation and checking are valid only for data lines that have their associated enable bits asserted.
DWE1:0#	O R(1) H(Q) P(Q)	**DRAM WRITE ENABLE** signals distinguish between read and write accesses to DRAM. DWE1:0# lines are asserted for writes and deasserted for reads. CAS7:0# determine valid bytes lanes during the access. These two outputs are functionally equivalent for all DRAM accesses; these provide increased drive capability for heavily loaded systems.
LEAF1:0#	O R(1) H(Q) P(Q)	**LEAF ENABLE** signals control the data output enables of the memory system during an interleaved DRAM read access. Use these to directly interface to either DRAM or transceiver output enable signals. LEAF0# is asserted during an even leaf access. LEAF1# is asserted during an odd leaf access.

Table 8. Memory Controller Signal Descriptions (Sheet 2 of 2)

NAME	TYPE	DESCRIPTION
MA11:0	O R(X) H(Q) P(Q)	**MULTIPLEXED ADDRESS** signals are multi-purpose depending on the device that is selected. For memory banks 0 and 1, these signals output address bits A13:2. These address bits are incremented for each data transfer of a burst access. For DRAM bank, these signals output the row/column multiplexed address bits 11:0. The relationship between the AD31:0 lines and the MA11:0 lines depends on the bank size, type and arrangement of the DRAM that is accessed.
MWE3:0#	O R(1) H(Q) P(Q)	**MEMORY WRITE ENABLE** signals for write accesses to SRAM/FLASH devices. The MWE's rising edge strobes valid data into these devices. MWE0# is asserted for writes to the BE0# lane MWE1# is asserted for writes to the BE1# lane MWE2# is asserted for writes to the BE2# lane MWE3# is asserted for writes to the BE3# lane
RAS3:0#	O R(1) H(Q) P(Q)	**ROW ADDRESS STROBE** signals are used for DRAM accesses and are asserted when the MA11:0 signals contain a valid row address. RAS3:0# always deasserts after the last data transfer in a DRAM access. *Non-Interleaved Operation:* RAS0# = Bank0 access RAS1# = Bank1 access RAS2# = Bank2 access RAS3# = Bank3 access *Interleaved Operation:* RAS0,2# = Bank0 access RAS1,3# = Bank1 access

Table 9. DMA, APIC, I^2C Units Signal Descriptions (Sheet 1 of 2)

NAME	TYPE	DESCRIPTION
DACK#	O R(1) H(Q) P(Q)	**DMA DEMAND MODE ACKNOWLEDGE** The DMA Controller asserts this signal to indicate (1) it can receive new data from an external device or (2) it has data to send to an external device.
DREQ#	I S(L)	**DMA DEMAND MODE REQUEST** External devices use this signal to indicate (1) new data is ready for transfer to the DMA controller or (2) buffers are available to receive data from the DMA controller.
PICCLK	I	**APIC BUS CLOCK** provides synchronous operation of the APIC bus.
PICD1:0	I/O OD R(Z) H(Q) P(Q)	**APIC DATA** lines comprise the data portion of the APIC 3-wire bus.

Table 9. DMA, APIC, I^2C Units Signal Descriptions (Sheet 2 of 2)

NAME	TYPE	DESCRIPTION
SCL	I/O OD R(Z) H(Q) P(Q)	**I^2C CLOCK** provides synchronous I^2C bus operation.
SDA	I/O OD R(Z) H(Q) P(Q)	**I^2C DATA** used for data transfer and arbitration on the I^2C bus.
WAIT#	O R(1) H(Q) P(Q)	**WAIT** is an output that allows the DMA controller to insert wait states during DMA accesses to an external memory system.

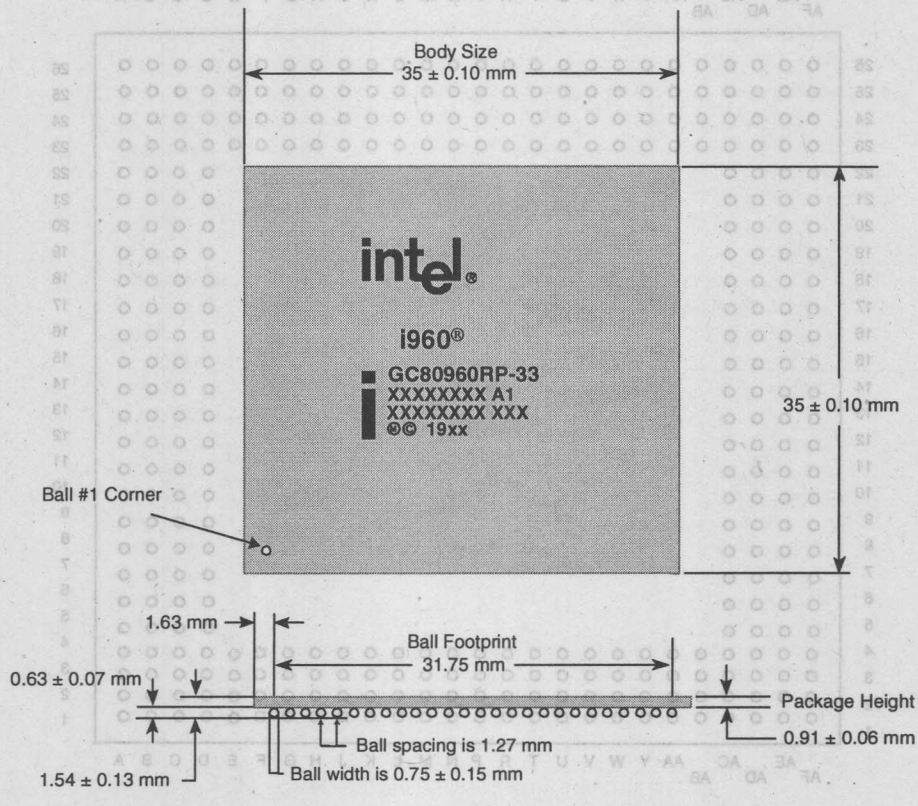
Table 10. Clock Signal

NAME	TYPE	DESCRIPTION
S_CLK	I	**SYNCHRONOUS PCI BUS CLOCK** Provides the processor's fundamental time base. All input/output timings are relative to S_CLK.

Table 11. ICE Signal Descriptions

NAME	TYPE	DESCRIPTION
ICEADS#	O	**ICE ADDRESS/DATA STATUS** indicates a valid address and the start of a new bus access. ICEADS# is active for accesses to external microcode.
ICEBRK#	I	**ICE BREAK** forces the processor to transition from emulation to interrogation mode.
ICEBUS7:0	I/O	**ICE BUS** is a bidirectional 8-bit bus linking the processor and the emulator. Used in various modes.
ICECLK	O	**ICE CLOCK** output signal to which all ICE bus signals are synchronized.
ICELOCK#	I	**ICE LOCK** is sampled during 80960 core reset to protect ICE configuration.
ICEMSG#	I	**ICE MESSAGE** signal used to acknowledge data from the processor to the emulator. Used only during interrogation mode.
ICESEL#	I	**ICESEL** enables or disables the ICE unit.
ICEVLD#	O	**ICE VALID** indicates the processor is driving the ICEBUS with valid data.
MSGFRM#	O	**ICE MESSAGE FRAME** indicates that trace messages are being issued to the ICEBUS. Used in emulation mode only.

3.1.2 352-Lead HL-PBGA Package



NOTES:
1. All dimensions and tolerances conform to ANSI Y14.5M 1982.
2. All dimensions are in millimeters.
3. Tin/Lead mix: 63%/37%.
4. Pad plating method: Electrolytic.
5. Encapsulant size: 22.38 x 22.38 mm (max).

Figure 3. 352L HL-PBGA Package Diagram (Top and Side View)

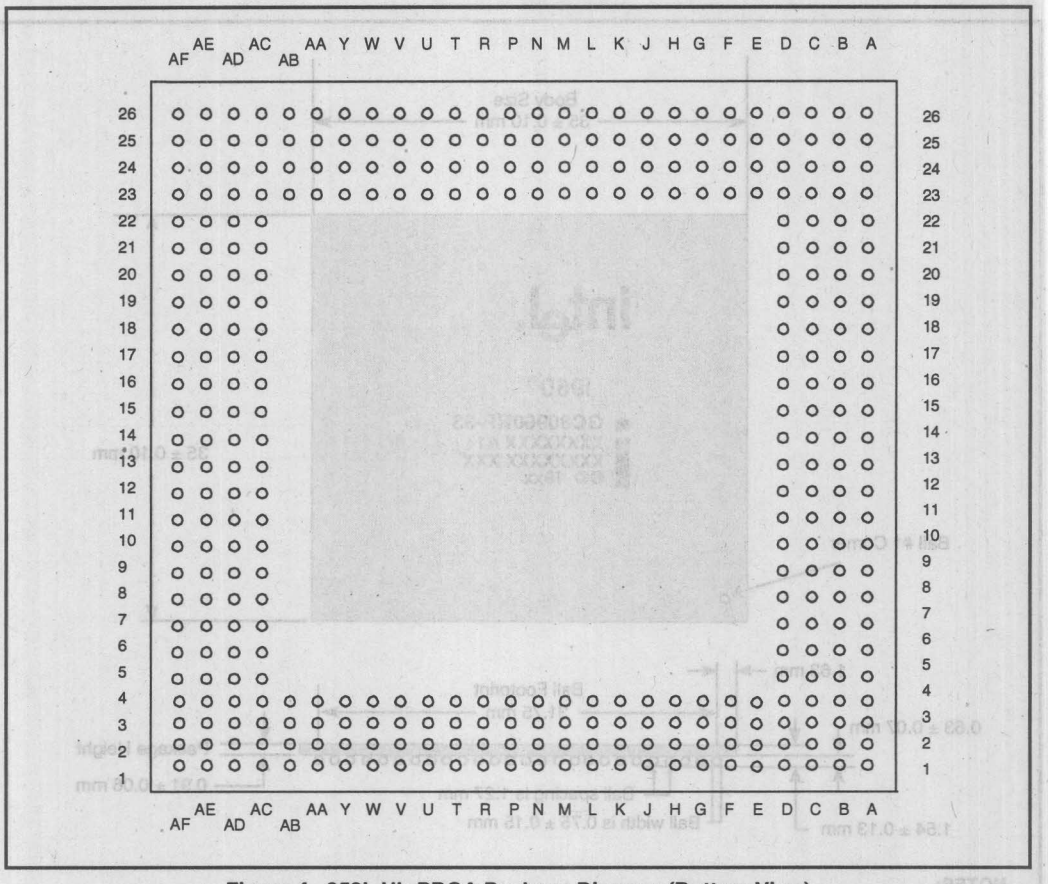


Figure 4. 352L HL-PBGA Package Diagram (Bottom View)

Table 12. 352-Lead HL-PBGA Package — Signal Name Order (Sheet 1 of 4)

Signal	Ball #	Signal	Ball #	Signal	Ball #
AD0	A18	CAS0#	F1	ICESEL#	AC2
AD1	B18	CAS1#	F2	ICEVLD#	AB1
AD2	C17	CAS2#	G3	LEAF0	L2
AD3	A17	CAS3#	G1	LEAF1	M3
AD4	B17	CAS4#	G2	LOCK#/ONCE#	AD4
AD5	C16	CAS5#	H3	LRDYRCV#	C19
AD6	A16	CAS6#	H1	LRST#	AD6
AD7	B16	CAS7#	H2	MA0	C7
AD8	C15	CE0#	L3	MA1	A7
AD9	A15	CE1#	L1	MA2	B7
AD10	B15	D/C#/RST_MODE#	AF4	MA3	C6
AD11	C14	DACK#	AD3	MA4	B6
AD12	A14	DALE0	M1	MA5	C5
AD13	B14	DALE1	M2	MA6	A5
AD14	C13	DEN#	A23	MA7	B5
AD15	A13	DP0	C2	MA8	C4
AD16	B13	DP1	D3	MA9	B4
AD17	C12	DP2	D1	MA10	C3
AD18	A12	DP3	D2	MA11	B3
AD19	B12	DREQ#	AD2	MSGFRM#	AC3
AD20	C11	DT/R#	B23	MWE0#	J3
AD21	A11	DWE0#	K1	MWE1#	J1
AD22	B11	DWE1#	K2	MWE2#	J2
AD23	C10	FAIL#	AD5	MWE3#	K3
AD24	A10	HOLD	V1	NC	A20
AD25	B10	HOLDA	V3	NC	AE4
AD26	C9	ICEADS#	AC1	NC	B20
AD27	A9	ICEBRK#	AA1	NC	C18
AD28	B9	ICEBUS0	V2	NMI#	T3
AD29	C8	ICEBUS1	W3	P_AD0	AD24
AD30	A8	ICEBUS2	W1	P_AD1	AE23
AD31	B8	ICEBUS3	W2	P_AD2	AF23
ADS#	B21	ICEBUS4	Y3	P_AD3	AD23
ALE	C20	ICEBUS5	Y1	P_AD4	AE22
BE0#	A22	ICEBUS6	Y2	P_AD5	AF22
BE1#	B22	ICEBUS7	AA3	P_AD6	AD22
BE2#	C21	ICECLK	AB2	P_AD7	AE21
BE3#	A21	ICELOCK#	AB3	P_AD8	AD21
BLAST#	C23	ICEMSG#	AA2	P_AD9	AE20

Table 12. 352-Lead HL-PBGA Package — Signal Name Order (Sheet 2 of 4)

Signal	Ball #
P_AD10	AF20
P_AD11	AD20
P_AD12	AE19
P_AD13	AF19
P_AD14	AD19
P_AD15	AE18
P_AD16	AE14
P_AD17	AF14
P_AD18	AD14
P_AD19	AE13
P_AD20	AF13
P_AD21	AD13
P_AD22	AE12
P_AD23	AF12
P_AD24	AF11
P_AD25	AD11
P_AD26	AE10
P_AD27	AF10
P_AD28	AD10
P_AD29	AE9
P_AD30	AF9
P_AD31	AD9
P_C/BE0#	AF21
P_C/BE1#	AF18
P_C/BE2#	AD15
P_C/BE3#	AE11
P_DEVSEL#	AF16
P_FRAME#	AF15
P_GNT#	AF8
P_IDSEL	AD12
P_INTA#	AF6
P_INTB#	AE6
P_INTC#	AD7
P_INTD#	AF7
P_IRDY#	AE15
P_LOCK#	AD17
P_PAR	AD18
P_PERR#	AF17
P_REQ#	AE8

Signal	Ball #
P_RST#	AE7
P_SERR#	AE17
P_STOP#	AE16
P_TRDY#	AD16
PICCLK	U3
PICD0	T1
PICD1	T2
RAS0#	E3
RAS1#	E1
RAS2#	E2
RAS3#	F3
RDYRCV#	B19
S_AD0	AE24
S_AD1	AD25
S_AD2	AC24
S_AD3	AC26
S_AD4	AC25
S_AD5	AB24
S_AD6	AB26
S_AD7	AB25
S_AD8	AA26
S_AD9	AA25
S_AD10	Y24
S_AD11	Y26
S_AD12	Y25
S_AD13	W24
S_AD14	W26
S_AD15	W25
S_AD16	R25
S_AD17	P24
S_AD18	P26
S_AD19	P25
S_AD20	N24
S_AD21	N26
S_AD22	N25
S_AD23	M24
S_AD24	L24
S_AD25	L26
S_AD26	L25

Signal	Ball #
S_AD27	K24
S_AD28	K26
S_AD29	K25
S_AD30	J24
S_AD31	J26
S_C/BE0#	AA24
S_C/BE1#	V24
S_C/BE2#	R26
S_C/BE3#	M25
S_CLK	F25
S_DEVSEL#	T24
S_FRAME#	R24
S_GNT0#/S_REQ#	H26
S_GNT1#	G24
S_GNT2#	G25
S_GNT3#	F26
S_GNT4#	E26
S_GNT5#	D24
S_IDSEL	M26
S_INTA#/XINT0#	N1
S_INTB#/XINT1#	N2
S_INTC#/XINT2#	P3
S_INTD#/XINT3#	P1
S_IRDY#	T25
S_LOCK#	U26
S_PAR	V26
S_PERR#	U24
S_REQ0#/S_GNT#	H24
S_REQ1#	H25
S_REQ2#	G26
S_REQ3#	F24
S_REQ4#	E24
S_REQ5#/S_ARB_EN	E25
S_RST#	J25
S_SERR#	V25
S_STOP#	U25
S_TRDY#	T26
SCL	U1
SDA	U2

V_{CC}	A3	V_{CC}	D11	V_{SS}	AC23
V_{CC}	A24	V_{CC}	D13	V_{SS}	D4
V_{CC}	A25	V_{CC}	D15	V_{SS}	D6
V_{CC}	A26	V_{CC}	D17	V_{SS}	D8
V_{CC}	AA23	V_{CC}	D19	V_{SS}	D10
V_{CC}	AB4	V_{CC}	D21	V_{SS}	D12
V_{CC}	AC6	V_{CC}	E23	V_{SS}	D14
V_{CC}	AC8	V_{CC}	F4	V_{SS}	D16
V_{CC}	AC10	V_{CC}	G23	V_{SS}	D18
V_{CC}	AC12	V_{CC}	H4	V_{SS}	D20
V_{CC}	AC14	V_{CC}	J23	V_{SS}	D22
V_{CC}	AC16	V_{CC}	K4	V_{SS}	D23
V_{CC}	AC18	V_{CC}	L23	V_{SS}	E4
V_{CC}	AC20	V_{CC}	M4	V_{SS}	F23
V_{CC}	AC22	V_{CC}	N23	V_{SS}	G4
V_{CC}	AD1	V_{CC}	P4	V_{SS}	H23
V_{CC}/V_{SS} (1)	AD8	V_{CC}	R23	V_{SS}	J4
V_{CC}	AD26	V_{CC}	T4	V_{SS}	K23
V_{CC}	AE1	V_{CC}	U23	V_{SS}	L4
V_{CC}	AE2	V_{CC}	V4	V_{SS}	M23
V_{CC}	AE25	V_{CC}	W23	V_{SS}	N4
V_{CC}	AE26	V_{CC}	Y4	V_{SS}	P23
V_{CC}	AF1	VCCPLL1	A19	V_{SS}	R4
V_{CC}	AF2	VCCPLL2	A6	V_{SS}	T23
V_{CC}	AF3	VCCPLL3	A4	V_{SS}	U4
V_{CC}	AF24	V_{SS}	AA4	V_{SS}	V23
V_{CC}	AF25	V_{SS}	AB23	V_{SS}	W4
V_{CC}	AF26	V_{SS}	AC4	V_{SS}	Y23
V_{CC}	B1	V_{SS}	AC5	W/R#	C22

NOTES:

1. Ball AD8 must be tied to either V_{CC} or V_{SS}.

Table 12. 352-Lead HL-PBGA Package — Signal Name Order (Sheet 4 of 4)

Signal	Ball #
WAIT#	N3
WIDTH/HLTD0	AF5
WIDTH/HLTD1/RETRY	AE5

Signal	Ball #
XINT4#	P2
XINT5#	R3
XINT6#	R1

Signal	Ball #
XINT7#	R2

Table 13. 352-Lead HL-PBGA Pinout — Ballpad Number Order (Sheet 1 of 4)

Ball #	Signal
A1	V_{CC}
A2	V_{CC}
A3	V_{CC}
A4	V_{CCPLL3}
A5	MA6
A6	V_{CCPLL2}
A7	MA1
A8	AD30
A9	AD27
A10	AD24
A11	AD21
A12	AD18
A13	AD15
A14	AD12
A15	AD9
A16	AD6
A17	AD3
A18	AD0
A19	V_{CCPLL1}
A20	NC
A21	BE3#
A22	BE0#
A23	DEN#
A24	V_{CC}
A25	V_{CC}
A26	V_{CC}
B1	V_{CC}
B2	V_{CC}
B3	MA11
B4	MA9
B5	MA7
B6	MA4
B7	MA2
B8	AD31
B9	AD28

Ball #	Signal
B10	AD25
B11	AD22
B12	AD19
B13	AD16
B14	AD13
B15	AD10
B16	AD7
B17	AD4
B18	AD1
B19	RDYRCV#
B20	NC
B21	ADS#
B22	BE1#
B23	DT/R#
B24	TCK
B25	V_{CC}
B26	V_{CC}
C1	V_{CC}
C2	DP0
C3	MA10
C4	MA8
C5	MA5
C6	MA3
C7	MA0
C8	AD29
C9	AD26
C10	AD23
C11	AD20
C12	AD17
C13	AD14
C14	AD11
C15	AD8
C16	AD5
C17	AD2
C18	NC

Ball #	Signal
C19	LRDYRCV#
C20	ALE
C21	BE2#
C22	W/R#
C23	BLAST#
C24	TMS
C25	TRST#
C26	V_{CC}
D1	DP2
D2	DP3
D3	DP1
D4	V_{SS}
D5	V_{CC}
D6	V_{SS}
D7	V_{CC}
D8	V_{SS}
D9	V_{CC}
D10	V_{SS}
D11	V_{CC}
D12	V_{SS}
D13	V_{CC}
D14	V_{SS}
D15	V_{CC}
D16	V_{SS}
D17	V_{CC}
D18	V_{SS}
D19	V_{CC}
D20	V_{SS}
D21	V_{CC}
D22	V_{SS}
D23	V_{SS}
D24	S_GNT5#
D25	TDO
D26	TDI
E1	RAS1#

Table 13. 352-Lead HL-PBGA Pinout — Ballpad Number Order (Sheet 2 of 4)

Ball #	Signal
E2	RAS2#
E3	RAS0#
E4	V_{SS}
E23	V_{CC}
E24	S_REQ4#
E25	S_REQ5#/S_ARB_EN
E26	S_GNT4#
F1	CAS0#
F2	CAS1#
F3	RAS3#
F4	V_{CC}
F23	V_{SS}
F24	S_REQ3#
F25	S_CLK
F26	S_GNT3#
G1	CAS3#
G2	CAS4#
G3	CAS2#
G4	V_{SS}
G23	V_{CC}
G24	S_GNT1#
G25	S_GNT2#
G26	S_REQ2#
H1	CAS6#
H2	CAS7#
H3	CAS5#
H4	V_{CC}
H23	V_{SS}
H24	S_REQ0#/S_GNT#
H25	S_REQ1#
H26	S_GNT0#/S_REQ#
J1	MWE1#
J2	MWE2#
J3	MWE0#
J4	V_{SS}
J23	V_{CC}
J24	S_AD30
J25	S_RST#
J26	S_AD31
K1	DWE0#
K2	DWE1#
K3	MWE3#
K4	V_{CC}
K23	V_{SS}
K24	S_AD27
K25	S_AD29
K26	S_AD28
L1	CE1#
L2	LEAF0#
L3	CE0#
L4	V_{SS}
L23	V_{CC}
L24	S_AD24
L25	S_AD26
L26	S_AD25
M1	DALE0
M2	DALE1
M3	LEAF1#
M4	V_{CC}
M23	V_{SS}
M24	S_AD23
M25	S_C/BE3#
M26	S_IDSEL
N1	S_INTA#/XINT0#
N2	S_INTB#/XINT1#
N3	WAIT#
N4	V_{SS}
N23	V_{CC}
N24	S_AD20
N25	S_AD22
N26	S_AD21
P1	S_INTD#/XINT3#
P2	XINT4#
P3	S_INTC#/XINT2#
P4	V_{CC}
P23	V_{SS}
P24	S_AD17
P25	S_AD19
P26	S_AD18
R1	XINT6#
R2	XINT7#
R3	XINT5#
R4	V_{SS}
R23	V_{CC}
R24	S_FRAME#
R25	S_AD16
R26	S_C/BE2#
T1	PICD0
T2	PICD1
T3	NMI#
T4	V_{CC}
T23	V_{SS}
T24	S_DEVSEL#
T25	S_IRDY#
T26	S_TRDY#
U1	SCL
U2	SDA
U3	PICCLK
U4	V_{SS}
U23	V_{CC}
U24	S_PERR#
U25	S_STOP#
U26	S_LOCK#
V1	HOLD
V2	ICEBUS0

Table 13. 352-Lead HL-PBGA Pinout — Ballpad Number Order (Sheet 3 of 4)

Ball #	Signal
V3	HOLDA
V4	V_{CC}
V23	V_{SS}
V24	S_C/BE1#
V25	S_SERR#
V26	S_PAR
W1	ICEBUS2
W2	ICEBUS3
W3	ICEBUS1
W4	V_{SS}
W23	V_{CC}
W24	S_AD13
W25	S_AD15
W26	S_AD14
Y1	ICEBUS5
Y2	ICEBUS6
Y3	ICEBUS4
Y4	V_{CC}
Y23	V_{SS}
Y24	S_AD10
Y25	S_AD12
Y26	S_AD11
AA1	ICEBRK#
AA2	ICEMSG#
AA3	ICEBUS7
AA4	V_{SS}
AA23	V_{CC}
AA24	S_C/BE0#
AA25	S_AD9
AA26	S_AD8
AB1	ICEVLD#
AB2	ICECLK
AB3	ICELOCK#

Ball #	Signal
AB4	V_{CC}
AB23	V_{SS}
AB24	S_AD5
AB25	S_AD7
AB26	S_AD6
AC1	ICEADS#
AC2	ICESEL#
AC3	MSGFRM#
AC4	V_{SS}
AC5	V_{SS}
AC6	V_{CC}
AC7	V_{SS}
AC8	V_{CC}
AC9	V_{SS}
AC10	V_{CC}
AC11	V_{SS}
AC12	V_{CC}
AC13	V_{SS}
AC14	V_{CC}
AC15	V_{SS}
AC16	V_{CC}
AC17	V_{SS}
AC18	V_{CC}
AC19	V_{SS}
AC20	V_{CC}
AC21	V_{SS}
AC22	V_{CC}
AC23	V_{SS}
AC24	S_AD2
AC25	S_AD4
AC26	S_AD3
AD1	V_{CC}
AD2	DREQ#

Ball #	Signal
AD3	DACK#
AD4	LOCK#/ONCE#
AD5	FAIL#
AD6	LRST#
AD7	P_INTC#
AD8	V_{CC}/V_{SS} (1)
AD9	P_AD31
AD10	P_AD28
AD11	P_AD25
AD12	P_IDSEL
AD13	P_AD21
AD14	P_AD18
AD15	P_C/BE2#
AD16	P_TRDY#
AD17	P_LOCK#
AD18	P_PAR
AD19	P_AD14
AD20	P_AD11
AD21	P_AD8
AD22	P_AD6
AD23	P_AD3
AD24	P_AD0
AD25	S_AD1
AD26	V_{CC}
AE1	V_{CC}
AE2	V_{CC}
AE3	STEST
AE4	NC
AE5	WIDTH/HLTD1/RETRY
AE6	P_INTB#
AE7	P_RST#
AE8	P_REQ#
AE9	P_AD29

NOTES:

1. Ball AD8 must be tied to either V_{CC} or V_{SS}.

Table 13. 352-Lead HL-PBGA Pinout — Ballpad Number Order (Sheet 4 of 4)

Ball #	Signal
AE10	P_AD26
AE11	P_C/BE3#
AE12	P_AD22
AE13	P_AD19
AE14	P_AD16
AE15	P_IRDY#
AE16	P_STOP#
AE17	P_SERR#
AE18	P_AD15
AE19	P_AD12
AE20	P_AD9
AE21	P_AD7
AE22	P_AD4
AE23	P_AD1
AE24	S_AD0

Ball #	Signal
AE25	V_{CC}
AE26	V_{CC}
AF1	V_{CC}
AF2	V_{CC}
AF3	V_{CC}
AF4	D/C#/RST_MODE#
AF5	WIDTH/HLTD0
AF6	P_INTA#
AF7	P_INTD#
AF8	P_GNT#
AF9	P_AD30
AF10	P_AD27
AF11	P_AD24
AF12	P_AD23
AF13	P_AD20

Ball #	Signal
AF14	P_AD17
AF15	P_FRAME#
AF16	P_DEVSEL#
AF17	P_PERR#
AF18	P_C/BE1#
AF19	P_AD13
AF20	P_AD10
AF21	P_C/BE0#
AF22	P_AD5
AF23	P_AD2
AF24	V_{CC}
AF25	V_{CC}
AF26	V_{CC}

3.2 Package Thermal Specifications

The device is specified for operation when T_C (case temperature) is within the range of 0° C to 115° C. Case temperature may be measured in any environment to determine whether the processor is within specified operating range. Measure the case temperature at the center of the top surface, opposite the ballpad.

3.2.1 Thermal Specifications

This section defines the terms used for thermal analysis.

3.2.1.1 Ambient Temperature

Ambient temperature, T_A, is the temperature of the ambient air surrounding the package. In a system environment, ambient temperature is the temperature of the air upstream from the package.

3.2.1.2 Case Temperature

To ensure functionality and reliability, the device is specified for proper operation when the case temperature, T_C, is within the specified range in Table 16.

When measuring case temperature, attention to detail is required to ensure accuracy. If a thermocouple is used, calibrate it before taking measurements. Errors may result when the measured surface temperature is affected by the surrounding ambient air temperature. Such errors may be due to a poor thermal contact between thermocouple junction and the surface, heat loss by radiation, or conduction through thermocouple leads.

To minimize measurement errors:

- Use a 35 gauge K-type thermocouple or equivalent.
- Attach the thermocouple bead or junction to the package top surface at a location corresponding to the center of the die (Figure 5). The center of the die gives a more accurate measurement and less variation as the boundary condition changes.
- Attach the thermocouple bead or junction at a 90° angle by an adhesive bond (such as thermal epoxy or heat-tolerant tape) to the package top surface as shown in Figure 5. When a heat sink is attached, drill a hole through the heat sink to allow contact with the package above the center of the die. The hole diameter should be no larger than 3.8 mm as shown in Figure 6.

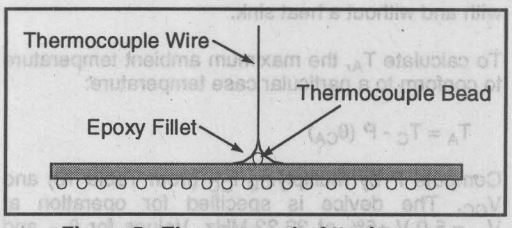


Figure 5. Thermocouple Attachment - No Heat Sink

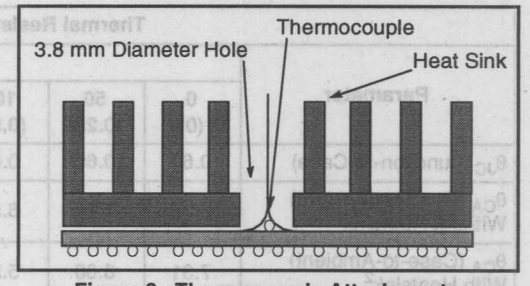


Figure 6. Thermocouple Attachment - With Heat Sink

3.2.1.3 Thermal Resistance

The thermal resistance value for the case-to-ambient, θ_{CA}, is used as a measure of the cooling solution's thermal performance.

3.2.2 Thermal Analysis

This thermal analysis is based on the following assumptions:

- Power dissipation is a constant 8 W.
- Maximum case temperature is 115° C.

Table 14 lists the case-to-ambient thermal resistances of the 80960RP for different air flow rates with and without a heat sink.

To calculate T_A, the maximum ambient temperature to conform to a particular case temperature:

$$T_A = T_C - P\ (\theta_{CA})$$

Compute P by multiplying I_{CC} (from Table 18) and V_{CC}. The device is specified for operation at V_{cc} = 5.0 V ±5% at 33.33 MHz. Values for θ_{JC} and θ_{CA} are given in Table 14.

Junction temperature (T_J) is commonly used in reliability calculations. T_J can be calculated from θ_{JC} (thermal resistance from junction to case) using the following equation:

$$T_J = T_C + P\ (\theta_{JC})$$

Similarly, when T_A is known, the corresponding case temperature (T_C) can be calculated as follows:

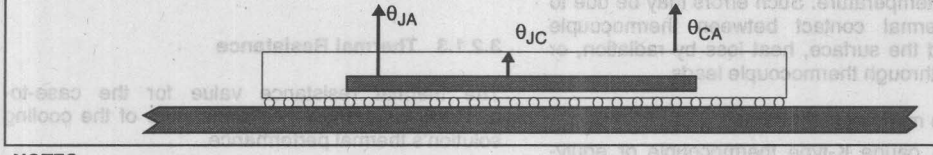
$$T_C = T_A + P\ (\theta_{CA})$$

The θ_{JA} (Junction to Ambient) for this package is currently estimated at 9.35° C/Watt with no airflow.

$$\theta_{JA} = \theta_{JC} + \theta_{CA}$$

Table 14. 352-Lead HL-PBGA Package Thermal Characteristics

Parameter	Thermal Resistance — °C/Watt							
	Airflow — ft./min (m/sec)							
	0 (0)	50 (0.25)	100 (0.50)	200 (1.01)	300 (1.52)	400 (2.03)	600 (3.04)	800 (4.06)
θ_{JC} (Junction-to-Case)	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60
θ_{CA} (Case-to-Ambient) Without Heatsink	8.75	7.42	6.58	6.11	5.79	5.61	5.49	5.47
θ_{CA} (Case-to-Ambient) With Heatsink[2]	7.31	6.00	5.21	4.80	4.52	4.37	4.26	4.23



NOTES:

1. This table applies to a HL-PBGA device soldered directly onto a board.
2. See Table 15 for heatsink information.

2021 West Valley View Lane Dallas, TX 75234-8993 Tel: (214) 243-4321 FAX: (214) 241-4656	BGA Clip: 20812-2		use with BGA Clip and Parker Chromerics Thermflow tape
Parker Chromerics 77 Dragon Court Woburn, MA 01888 Tel: (617) 935-4850 FAX: (617) 933-4318	T705	NA	Thermflow tape; use with Thermalloy BGA Clip
AAVID Thermal Technologies, Inc. One Kool Path P.O. Box 400 Laconia, N.H. 13247-0400 Tel: (603) 528-3400 FAX: (603) 527-2129	Heatsink: 364424B00032	40.5 X 40 X 11	AAVID Heatsink; use with pre-applied thermal adhesive tape (Ther-A-Grip)

4.0 ELECTRICAL SPECIFICATIONS

NOTICE: This data sheet contains information on products in the sampling and initial production phases of development. The specifications are subject to change without notice. Contact your local Intel representative before finalizing a design.

4.1 Absolute Maximum Ratings

Parameter	Maximum Rating
Storage Temperature	–55° C to + 125° C
Case Temperature Under Bias	0° C to + 115° C
Supply Voltage wrt. V_{SS}	–0.5 V to + 6.5 V
Voltage on Any Ball wrt. V_{SS}	–0.5 V to V_{CC} + 0.5 V

†**WARNING:** *Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

Table 16. Operating Conditions

Symbol	Parameter	Min	Max	Units	Notes
V_{CC}	Supply Voltage	4.75	5.25	V	
F_{S_CLK}	Input Clock Frequency	16	33.33	MHz	
T_C	Case Temperature Under Bias GC80960RP (352 HL-PBGA)	0	115	°C	

4.2 Targeted DC Specifications

Table 17. DC Characteristics

Symbol	Parameter	Min	Max	Units	Notes
V_{IL}	Input Low Voltage	-0.5	0.8	V	(1)
V_{IH}	Input High Voltage	2.0	V_{CC} + 0.5	V	(1)
V_{OL1}	Output Low Voltage Processor signals		0.45	V	I_{OL} = 6 mA (3)
V_{OH1}	Output High Voltage Processor signals	2.4 V_{CC} - 0.5		V	I_{OH} = -2 mA (3) I_{OH} = -200 µA (3)
V_{OL2}	Output Low Voltage PCI signals		0.55	V	I_{OL} = 6 mA (1)
V_{OH2}	Output High Voltage PCI signals	2.4		V	I_{OH} = -2 mA (1)
V_{OL3}	Output Low Voltage Memory Controller Normal drive		0.45	V	I_{OL} = 6 mA (4)
V_{OH3}	Output High Voltage Memory Controller Normal drive	2.4		V	I_{OH} = -2 mA (4)
V_{OL4}	Output Low Voltage Memory Controller High Drive		0.45	V	I_{OL} = 7 mA
V_{OH4}	Output High Voltage Memory Controller High Drive	2.4		V	I_{OH} = -2 mA
V_{OL5}	Output Low Voltage APIC Data Lines		0.45	V	I_{OL} = 20 mA
C_{IN}	Input Capacitance - HL-PBGA		10	pF	$F_{S\_CLK} = T_F$ Min (1, 2)
C_{OUT}	I/O or Output Capacitance - HL-PBGA		10	pF	$F_{S\_CLK} = T_F$ Min (1, 2)
C_{CLK}	S_CLK Capacitance - HL-PBGA	5	12	pF	$F_{S\_CLK} = T_F$ Min (1, 2)
C_{IDSEL}	IDSEL Ball Capacitance		8	pF	(1)
L_{PIN}	Ball Inductance		20	nH	(1)

NOTES:

1. As required by the *PCI Local Bus Specification* Revision 2.1.
2. Not tested.
3. Processor signals include AD31:0, ALE, ADS#, BE3:0#, WIDTH/HLTD0, WIDTH/HLTD1/RETRY, D/C#/RST_MODE#, W/R#, DT/R#, DEN#, BLAST#, LRDYRCV#, LOCK#/ONCE#, HOLD, FAIL#, TDO, DACK#, WAIT#, SDA, SCL.
4. Memory Controller signals include MA11:0, DP3:0, RAS3:0#, CAS7:0#, MWE3:0#, DWE1:0#, DALE1:0, CE1:0#, LEAF1:0#.
5. Memory Controller signals capable of high drive are MA11:0, CAS7:0#, RAS3:0#, DWE1:0#.

Table 18. I_{CC} Characteristics

Symbol	Parameter	Typ	Max	Units	Notes
I_{LI1}	Input Leakage Current for each signal except LOCK#/ONCE#, WIDTH/HLTD0, WIDTH/HLTD1/RETRY, BLAST#, D/C#/RST_MODE#, DEN#,TMS, TRST#, TDI		± 5	µA	$0 \le V_{IN} \le V_{CC}$
I_{LI2}	Input Leakage Current for LOCK#/ONCE#, WIDTH/HLTD0, WIDTH/HLTD1/RETRY, BLAST#, D/C#/RST_MODE#, DEN#, TMS, TRST#, TDI	-140	-250	µA	V_{IN} = 0.45 V (1)
I_{LO}	Output Leakage Current		± 5	µA	$0.4 \le V_{OUT} \le V_{CC}$
I_{CC} Active (Power Supply)	Power Supply Current		1.52	A	(1,2)
I_{CC} Active (Thermal)	Thermal Current	1.28		A	(1,3)
I_{CC} Active (Power Modes)	Reset Mode ONCE Mode		1.09 0.25	A	(4) (4)

NOTES:

1. Measured with device operating and outputs loaded to the test condition in Figure 7.
2. I_{CC} Active (Power Supply) value is provided for selecting your system's power supply. It is measured using one of the worst case instruction mixes with V_{CC} = 5.25 V and ambient temperature = 45 ° C. This parameter is characterized but not tested.
3. I_{CC} Active (Thermal) value is provided for your system's thermal management. Typical I_{CC} is measured with V_{CC} = 5.0 V and ambient temperature = 45 ° C. This parameter is characterized but not tested.
4. I_{CC} Test (Power modes) refers to the I_{CC} values that are tested when the device is in Reset mode or ONCE mode with V_{CC} = 5.25 V and ambient temperature = 45 ° C.

T_{CH}	S_CLK High Time	12		ns	Measured at 1.5 V (2,3)
T_{CL}	S_CLK Low Time	12		ns	Measured at 1.5 V (2,3)
T_{CR}	S_CLK Rise Time		4	V/ns	0.4 V to 2.4 V (2,3)
T_{CF}	S_CLK Fall Time		4	V/ns	2.4 V to 0.4 V (2,3)

NOTES:

1. See Figure 8.
2. To ensure a 1:1 relationship between the amplitude of the input jitter and the internal clock, the jitter frequency spectrum should not have any power peaking between 500 KHz and 1/3 of the S_CLK frequency.
3. Not tested.

Table 20. Synchronous Output Timings

Symbol	Parameter	Min	Max	Units	Notes
T_{OV1}	Output Valid Delay - All Signals Except ALE Inactive and DT/R#	3	15.5	ns	(1,2)
T_{OV2}	Output Valid Delay, DT/R#	0.5 T_C +3	0.5 T_C +15	ns	(2)
T_{OV3}	Output Valid Delay - PCI Signals Except P_REQ#, S_GNT0#/S_REQ#, and S_GNT5:1#	2	12	ns	(2)
T_{OV4}	Output Valid Delay P_REQ#, S_GNT0#/S_REQ#, and S_GNT5:1#	2	12	ns	(2)
T_{OV5}	Output Valid Delay - DP3:0	3	19	ns	(2)
T_{OF}	Output Float Delay	3	13	ns	(3,4)

NOTES:

1. Inactive ALE refers to the falling edge of ALE. For inactive ALE timings, see Table 22, Relative Output Timings (pg. 39).
2. See Figure 9, T_{OV} Output Delay Waveform (pg. 45).
3. A float condition occurs when the output current becomes less than I_{LO}. Float delay is not tested, but is designed to be no longer than the valid delay.
4. See Figure 10, T_{OF} Output Float Waveform (pg. 45).

Table 21. Synchronous Input Timings

Sym	Parameter	Min	Max	Units	Notes
T_{IS1}	Input Setup to S_CLK — AD31:0, NMI#, XINT7:4#, S_INT[A:D]#/XINT3:0#, DP3:0#	7		ns	(1,2,4)
T_{IH1}	Input Hold from S_CLK — AD31:0, NMI#, XINT7:4#, S_INT[A:D]#/XINT3:0#, DP3:0#	2		ns	(1,2)
T_{IS2}	Input Setup to S_CLK — RDYRCV# and HOLD	10		ns	(2)
T_{IH2}	Input Hold from S_CLK — RDYRCV# and HOLD	2		ns	(2)
T_{IS3}	Input Setup to S_CLK — LOCK#/ONCE#, STEST	7		ns	(1,2)
T_{IH3}	Input Hold from S_CLK — LOCK#/ONCE#, STEST	3		ns	(1,2)
T_{IS4}	Input Setup to S_CLK — DREQ#	12		ns	(2)
T_{IH4}	Input Hold from S_CLK — DREQ#	7		ns	(2)
T_{IS5}	Input Setup to S_CLK — PCI Signals Except P_GNT#, S_REQ0#/S_GNT#, and S_REQ5:1#	8		ns	(2,4)
T_{IH5}	Input Hold from S_CLK — PCI Signals	0		ns	(2,4)
T_{IS6}	Input Setup to S_CLK — P_RST#	6		ns	(2,3)
T_{IH6}	Input Hold to S_CLK — P_RST#	2		ns	(2,3)
T_{IS7}	Input Setup to S_CLK — P_GNT#	10		ns	(2)
T_{IS8}	Input Setup to S_CLK — S_REQ0#/S_GNT# and S_REQ5:1#	12		ns	(2)
T_{IS9}	Input Setup to P_RST# — WIDTH/HLTD0, WIDTH/HLTD1/RETRY, D/C#/RST_MODE#	7		ns	(1,2,4)
T_{IH9}	Input Hold from P_RST# — WIDTH/HLTD0, WIDTH/HLTD1/RETRY, D/C#/RST_MODE#	3		ns	(1,2,4)

NOTES:

1. Setup and hold times must be met for proper processor operation. NMI#, XINT7:4#, and S_INT[A:D]#/XINT3:0# may be synchronous or asynchronous. Meeting setup and hold time guarantees recognition at a particular clock edge. For asynchronous operation, NMI#, XINT7:4#, and S_INT[A:D]#/XINT3:0# must be asserted for a minimum of two S_CLK periods to guarantee recognition.
2. See Figure 11, T_{IS} and T_{IH} Input Setup and Hold Waveform (pg. 46).
3. P_RST# may be synchronous or asynchronous. Meeting setup and hold time guarantees recognition at a particular clock edge.
4. Guaranteed by design. May not be 100% tested.

4.3.1 Relative Output Timings

Table 22. Relative Output Timings

Symbol	Parameter	Min	Max	Units	Notes
T_{LXL}	ALE Width	$0.5T_C$-3		ns	(1, 2)
T_{LXA}	Address Hold from ALE Inactive	$0.5T_C$-3		ns	Equal Loading (1, 2)
T_{DXD}	DT/R# Valid to DEN# Active	$0.5T_C$-3		ns	Equal Loading (1, 3)

NOTES:
1. Guaranteed by design. May not be 100% tested.
2. See Figure 12.
3. See Figure 13.

4.3.2 Memory Controller Relative Output Timings

Table 23. Fast Page Mode Non-Interleaved DRAM Output Timings

Symbol	Description	Min	Max	Units	Notes
T_{OV6}	RAS3:0# Rising and Falling edge Output Valid Delay	2	10	ns	
T_{OV7}	CAS7:0# Rising Edge Output Valid Delay	2	10	ns	
T_{OV8}	CAS7:0# Falling Edge Output Valid Delay	0.5Tc+2	0.5Tc+10	ns	1
T_{OV9}	MA11:0 Output Valid Delay-Row Address	0.5Tc+2	0.5Tc+11	ns	1
T_{OV10}	MA11:0 Output Valid Delay-Column Address	2	12.5	ns	
T_{OV11}	DWE1:0# Rising and Falling edge Output Valid Delay	2	11	ns	

NOTES:
1. Signal generated on the rising edge of an internally generated 2XCLK which corresponds to the center of an S_CLK period. For testing purposes, the signal is specified relative to the rising edge of S_CLK with the 0.5Tc period offset.

Table 24. Fast Page Mode Interleaved DRAM Output Timings (Sheet 1 of 2)

Symbol	Description	Min	Max	Units	Notes
T_{OV12}	RAS3:0# Rising and Falling edge Output Valid Delay	2	10	ns	
T_{OV13}	CAS7:0# Rising Edge Output Valid Delay	2	10	ns	
T_{OV14}	CAS7:0# Falling Edge Output Valid Delay	0.5Tc+2	0.5Tc+10	ns	1
T_{OV15}	MA11:0 Output Valid Delay-Row Address	0.5Tc+2	0.5Tc+11	ns	1
T_{OV16}	MA11:0 Output Valid Delay-Column Address	2	12.5	ns	

NOTE:
1. Signal generated on the rising edge of an internally generated 2XCLK which corresponds to the center of an S_CLK period. For testing purposes, the signal is specified relative to the rising edge of S_CLK with the 0.5Tc period offset.

Table 24. Fast Page Mode Interleaved DRAM Output Timings (Sheet 2 of 2)

Symbol	Description	Min	Max	Units	Notes
T_{OV17}	DWE1:0# Rising and Falling Edge Output Valid Delay	2	11	ns	
T_{OV18}	DALE1:0 Initial Falling Edge Output Valid Delay	2	11	ns	
T_{OV19}	DALE1:0 Burst Falling Edge Output Valid Delay	0.5Tc+2	0.5Tc+11	ns	1
T_{OV20}	DALE1:0 Rising Edge Output Valid Delay	2	11	ns	
T_{OV21}	LEAF1:0# Rising and Falling Edge Output Valid Delay	2	11	ns	

NOTE:

1. Signal generated on the rising edge of an internally generated 2XCLK which corresponds to the center of an S_CLK period. For testing purposes, the signal is specified relative to the rising edge of S_CLK with the 0.5Tc period offset.

Table 25. EDO DRAM Output Timings

Symbol	Description	Min	Max	Units	Notes
T_{OV22}	RAS3:0# Rising and Falling Edge Output Valid Delay	2	10	ns	
T_{OV23}	CAS7:0# Rising Edge Output Valid Delay - Read Cycles	0.5Tc+2	0.5Tc+10	ns	1
T_{OV24}	CAS7:0# Falling Edge Output Valid Delay - Read Cycles	2	10	ns	
T_{OV25}	CAS7:0# Rising Edge Output Valid Delay - Write Cycles	2	10	ns	
T_{OV26}	CAS7:0# Falling Edge Output Valid Delay - Write Cycles	0.5Tc+2	0.5Tc+10	ns	1
T_{OV27}	MA11:0 Output Valid Delay - Row Address	0.5Tc+2	0.5Tc+11	ns	1
T_{OV28}	MA11:0 Output Valid Delay - Column Address Read Cycles	0.5Tc+2	0.5Tc+12.5	ns	1
T_{OV29}	MA11:0 Output Valid Delay - Column Address Write Cycles	2	12.5	ns	
T_{OV30}	DWE1:0# Rising and Falling Edge Output Valid Delay	2	11	ns	

NOTES:

1. Signal generated on the rising edge of an internally generated 2XCLK which corresponds to the center of an S_CLK period. For testing purposes, the signal is specified relative to the rising edge of S_CLK with the 0.5Tc period offset.

Table 26. BEDO DRAM Output Timings

Symbol	Description	Min	Max	Units	Notes
T_{OV31}	RAS3:0# Rising and Falling Edge Output Valid Delay	2	11	ns	
T_{OV32}	CAS7:0# Rising Edge Output Valid Delay - Read Cycles	0.5Tc+2	0.5Tc+11	ns	1
T_{OV33}	CAS7:0# Falling Edge Output Valid Delay - Read Cycles	2	11	ns	
T_{OV34}	CAS7:0# Rising Edge Output Valid Delay - Write Cycles	2	11	ns	
T_{OV35}	CAS7:0# Falling Edge Output Valid Delay - Write Cycles	0.5Tc+2	0.5Tc+11	ns	1
T_{OV36}	MA11:0 Output Valid Delay - Row Address	0.5Tc +2	0.5Tc+11	ns	1
T_{OV37}	MA11:0 Output Valid Delay - Column Address Read Cycles	0.5Tc +2	0.5Tc+14	ns	1
T_{OV38}	MA11:0 Output Valid Delay - Column Address Write Cycles	2	14	ns	
T_{OV39}	DWE1:0# Rising and Falling Edge Output Valid Delay	2	11	ns	

NOTES:
1. Signal generated on the rising edge of an internally generated 2XCLK which corresponds to the center of an S_CLK period. For testing purposes, the signal is specified relative to the rising edge of S_CLK with the 0.5Tc period offset.

Table 27. SRAM/ROM Output Timings

Symbol	Description	Min	Max	Units	Notes
T_{OV40}	CE1:0# Rising and Falling Edge Output Valid Delay	2	11	ns	
T_{OV41}	MWE3:0# Rising Edge Output Valid Delay	2	11	ns	
T_{OV42}	MWE3:0# Falling Edge Output Valid Delay	0.5Tc +1	0.5Tc +11	ns	1
T_{OV43}	MA11:0 Output Valid Delay - Initial Address	0.5Tc +2	0.5Tc +14	ns	
T_{OV44}	MA11:0 Output Valid Delay - Burst Address	2	14	ns	

NOTES:
1. Signal generated on the rising edge of an internally generated 2XCLK which corresponds to the center of an S_CLK period. For testing purposes, the signal is specified relative to the rising edge of S_CLK with the 0.5Tc period offset.

4.3.3 Boundary Scan Test Signal Timings

Table 28. Boundary Scan Test Signal Timings

Symbol	Parameter	Min	Max	Units	Notes
T_{BSF}	TCK Frequency	0	$.5T_F$	MHz	
T_{BSCH}	TCK High Time	15		ns	Measured at 1.5 V (1)
T_{BSCL}	TCK Low Time	15		ns	Measured at 1.5 V (1)
T_{BSCR}	TCK Rise Time		5	ns	0.8 V to 2.0 V (1)
T_{BSCF}	TCK Fall Time		5	ns	2.0 V to 0.8 V (1)
T_{BSIS1}	Input Setup to TCK — TDI, TMS	4		ns	
T_{BSIH1}	Input Hold from TCK — TDI, TMS	6		ns	
T_{BSOV1}	TDO Valid Delay	3	30	ns	Relative to falling edge of TCK
T_{BSOF1}	TDO Float Delay	3	30	ns	Relative to falling edge of TCK
T_{BSOV2}	All Outputs (Non-Test) Valid Delay	3	30	ns	Relative to falling edge of TCK
T_{BSOF2}	All Outputs (Non-Test) Float Delay	3	30	ns	Relative to falling edge of TCK
T_{BSIS2}	Input Setup to TCK — All Inputs (Non-Test)	4		ns	
T_{BSIH2}	Input Hold from TCK — All Inputs (Non-Test)	6		ns	

NOTES:
1. Not tested.

4.3.4 APIC Bus Interface Signal Timings

Table 29. APIC Bus Interface Signal Timings

Symbol	Parameter	Min	Max	Units	Notes
T_{APF}	PICCLK Frequency	2	16.66	MHz	
T_{APC}	PICCLK Period	60	500	ns	
T_{APCH}	PICCLK High Time	9		ns	
T_{APCL}	PICCLK Low Time	9		ns	
T_{APCR}	PICCLK Rise Time	1	5	ns	(1)
T_{APCF}	PICCLK Fall Time	1	5	ns	(1)
T_{APIS1}	Input Setup to PICCLK — PICD1:0	3		ns	
T_{APIH1}	Input Hold from PICCLK — PICD1:0	2.5		ns	
T_{APOF}	Output Float Delay from PICCLK — PICD1:0	4	16	ns	(1)
T_{APOVI}	Output Valid Delay from PICCLK — PICD1:0 (High to Low)	4	22	ns	

NOTES:
1. Not tested.

4.3.5 I^2C Interface Signal Timings

Table 30. I^2C Interface Signal Timings

Symbol	Parameter	Std. Mode		Fast Mode		Units	Notes
		Min	Max	Min	Max		
F_{SCL}	SCL Clock Frequency	0	100	0	400	KHz	
T_{BUF}	Bus Free Time Between STOP and START Condition	4.7		1.3		µs	(1)
T_{HDSTA}	Hold Time (repeated) START Condition	4		0.6		µs	(1,3)
T_{LOW}	SCL Clock Low Time	4.7		1.3		µs	(1,2)
T_{HIGH}	SCL Clock High Time	4		0.6		µs	(1,2)
T_{SUSTA}	Setup Time for a Repeated START Condition	4.7		0.6		µs	(1)
T_{HDDAT}	Data Hold Time	0		0	0.9	µs	(1)
T_{SUDAT}	Data Setup Time	250		100		ns	(1)
T_R	SCL and SDA Rise Time		1000	$20+0.1C_b$	300	ns	(1,4)
T_F	SCL and SDA Fall Time		300	$20+0.1C_b$	300	ns	(1,4)
T_{SUSTO}	Setup Time for STOP Condition	4		0.6		µs	(1)

NOTES:

1. See Figure 14.
2. Not tested.
3. After this period, the first clock pulse is generated.
4. C_b = the total capacitance of one bus line, in pF.

4.4 AC Test Conditions

The AC Specifications in Section 4.3, Targeted AC Specifications (pg. 37) are tested with the 50 pF load indicated in Figure 7.

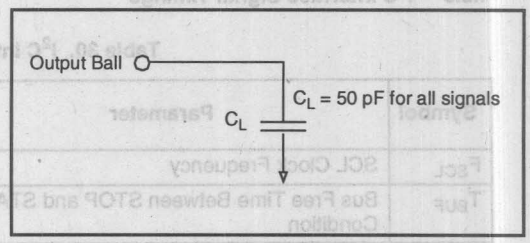


Figure 7. AC Test Load

4.5 AC Timing Waveforms

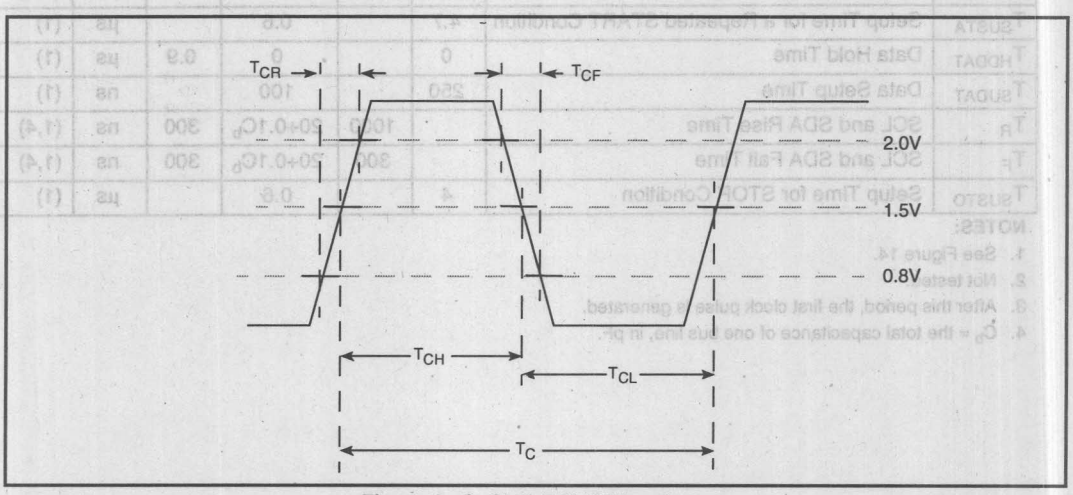


Figure 8. S_CLK, TCLK Waveform

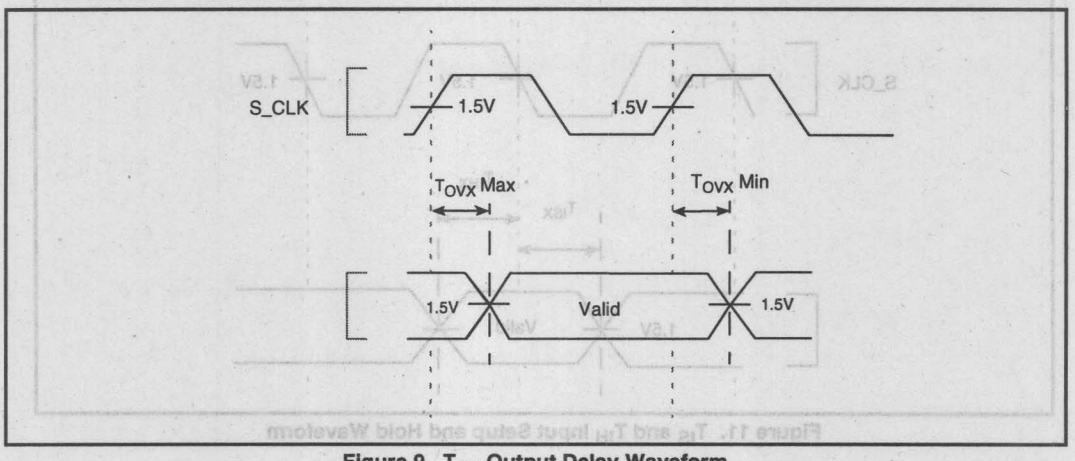


Figure 9. T_{OV} Output Delay Waveform

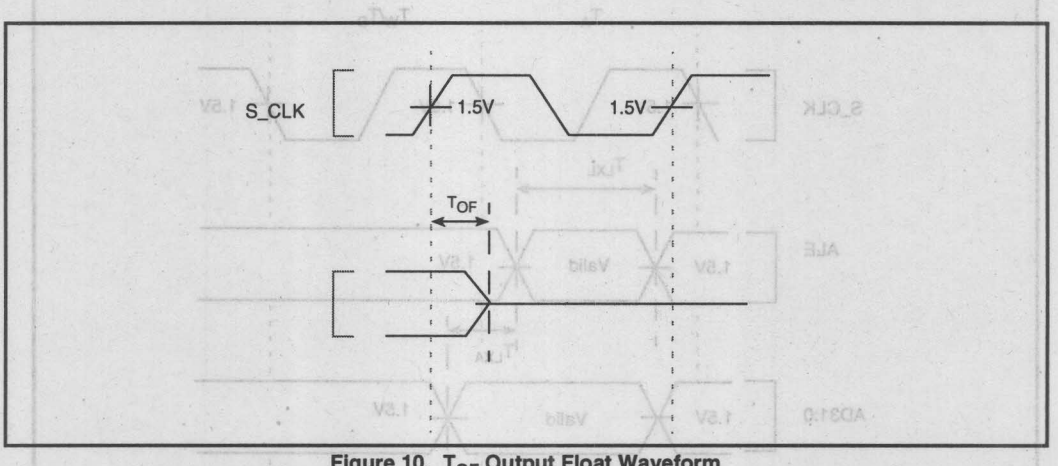


Figure 10. T_{OF} Output Float Waveform

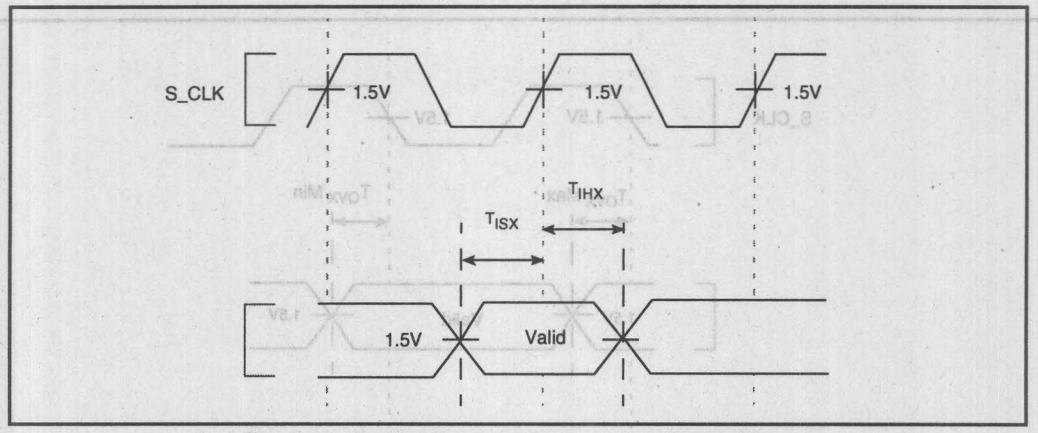


Figure 11. T_{IS} and T_{IH} Input Setup and Hold Waveform

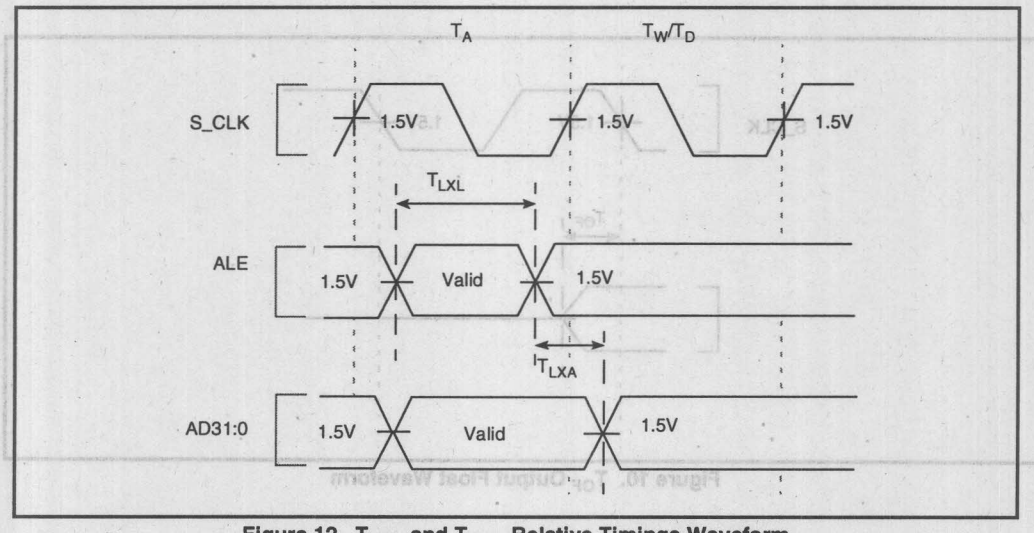


Figure 12. T_{LXL} and T_{LXA} Relative Timings Waveform

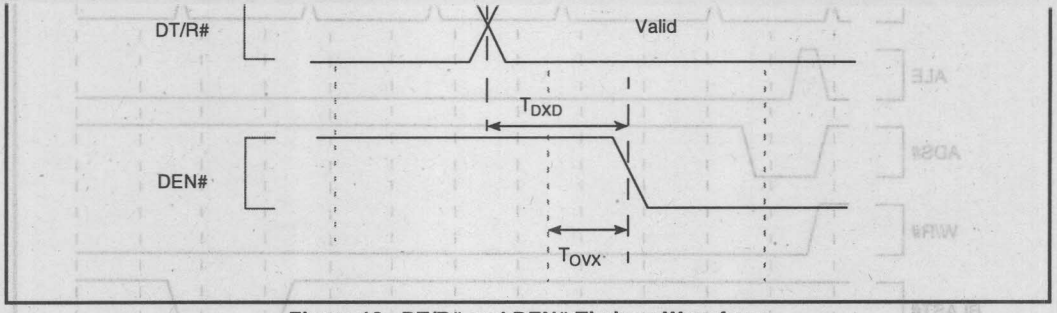


Figure 13. DT/R# and DEN# Timings Waveform

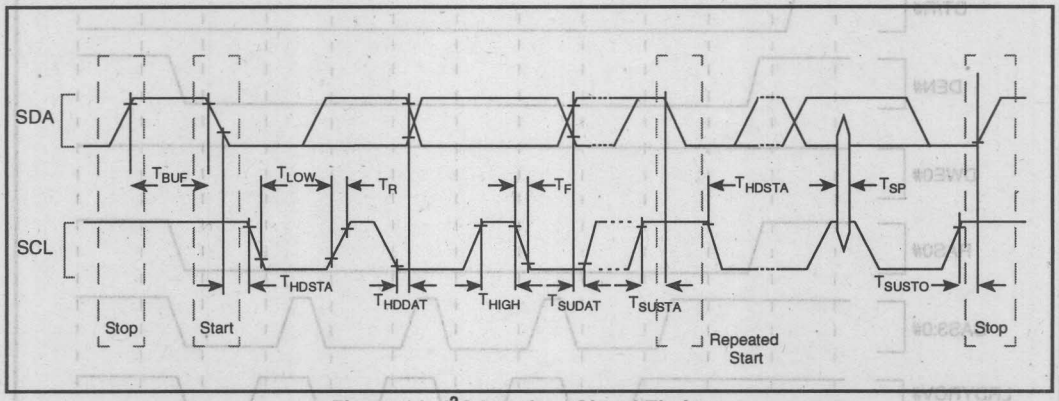


Figure 14. I^2C Interface Signal Timings

4.6 Memory Controller Output Timing Waveforms

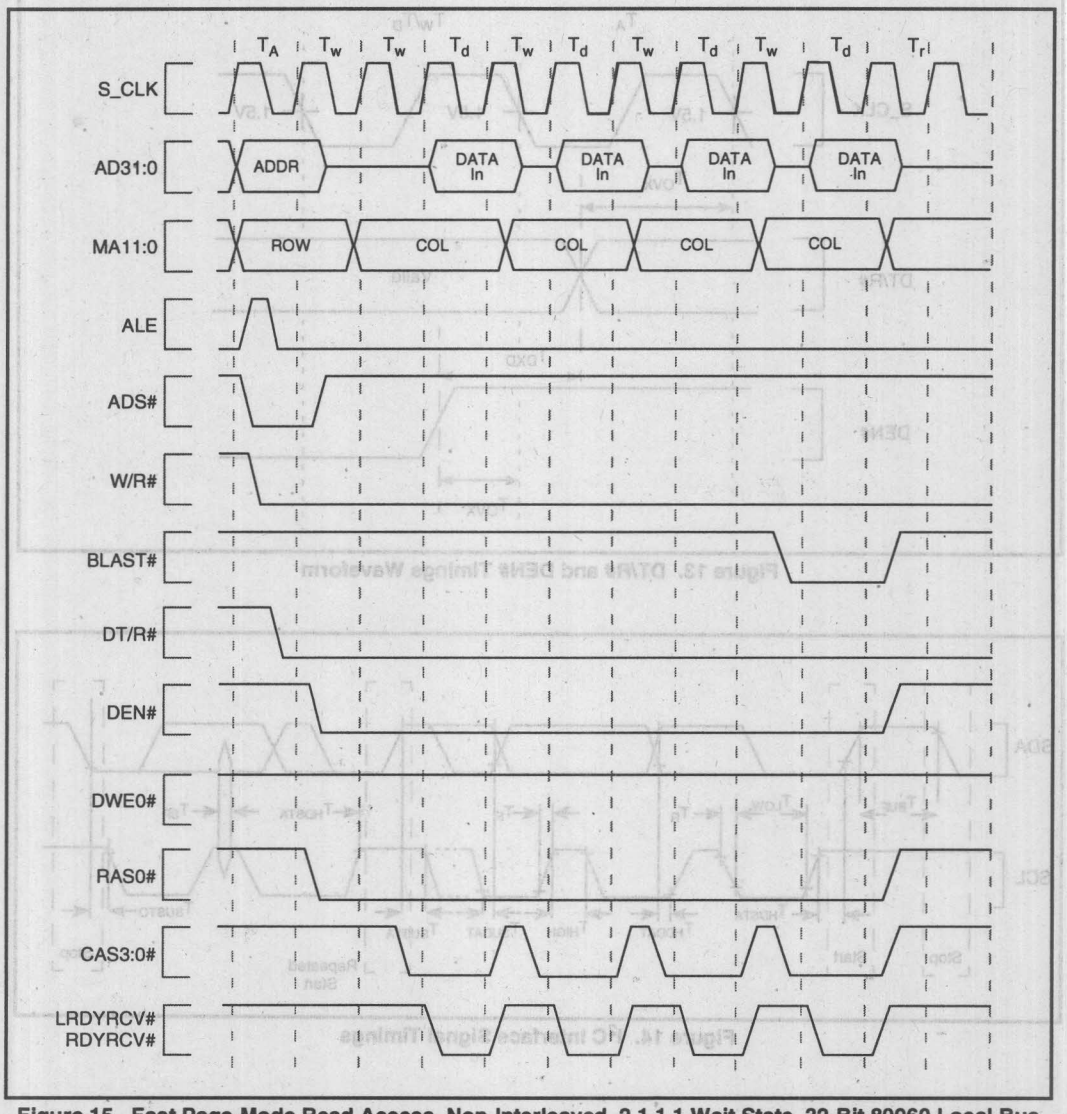


Figure 15. Fast Page-Mode Read Access, Non-Interleaved, 2,1,1,1 Wait State, 32-Bit 80960 Local Bus

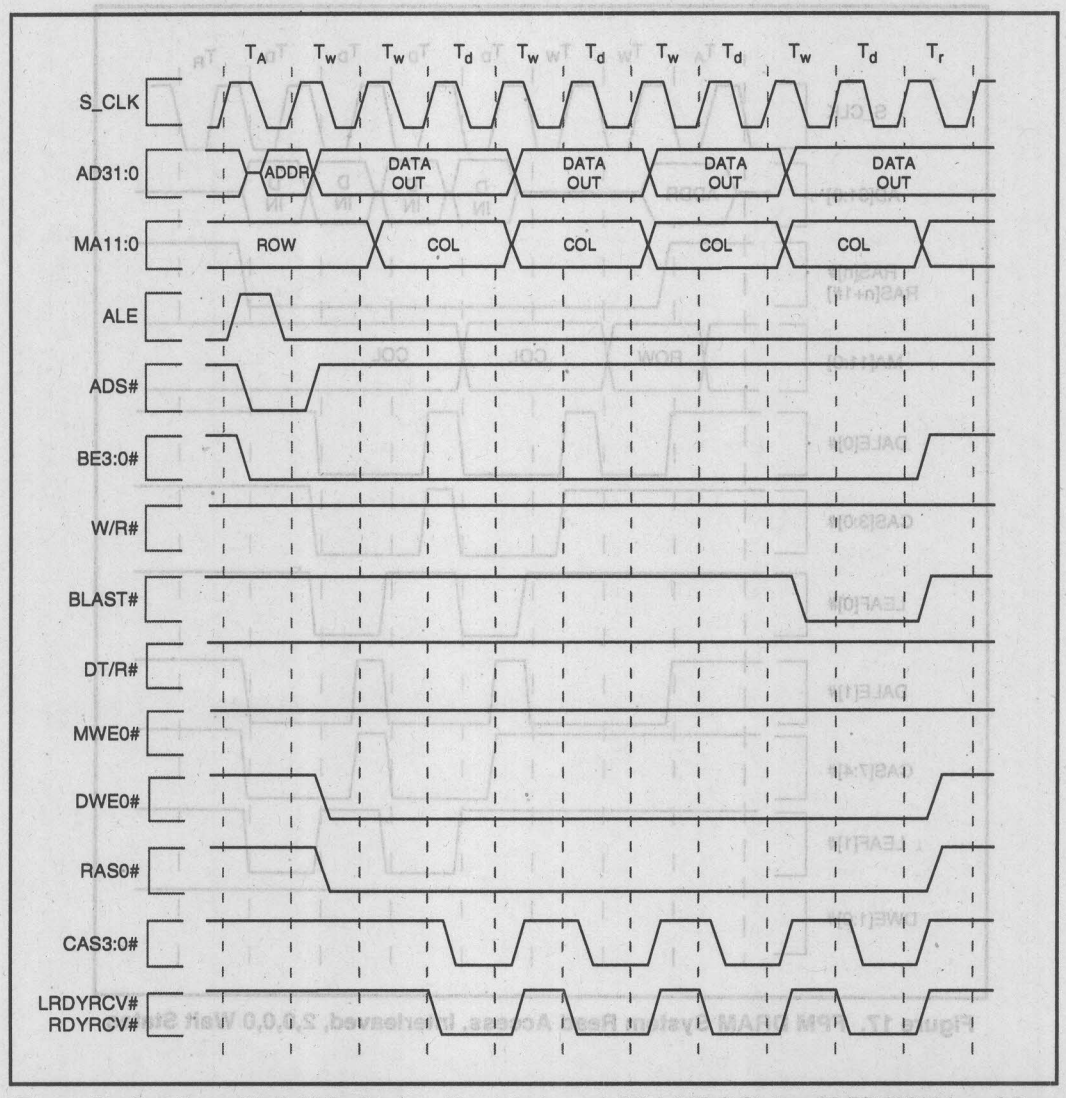


Figure 16. Fast Page-Mode Write Access, Non-Interleaved, 2,1,1,1 Wait States, 32-Bit 80960 Local Bus

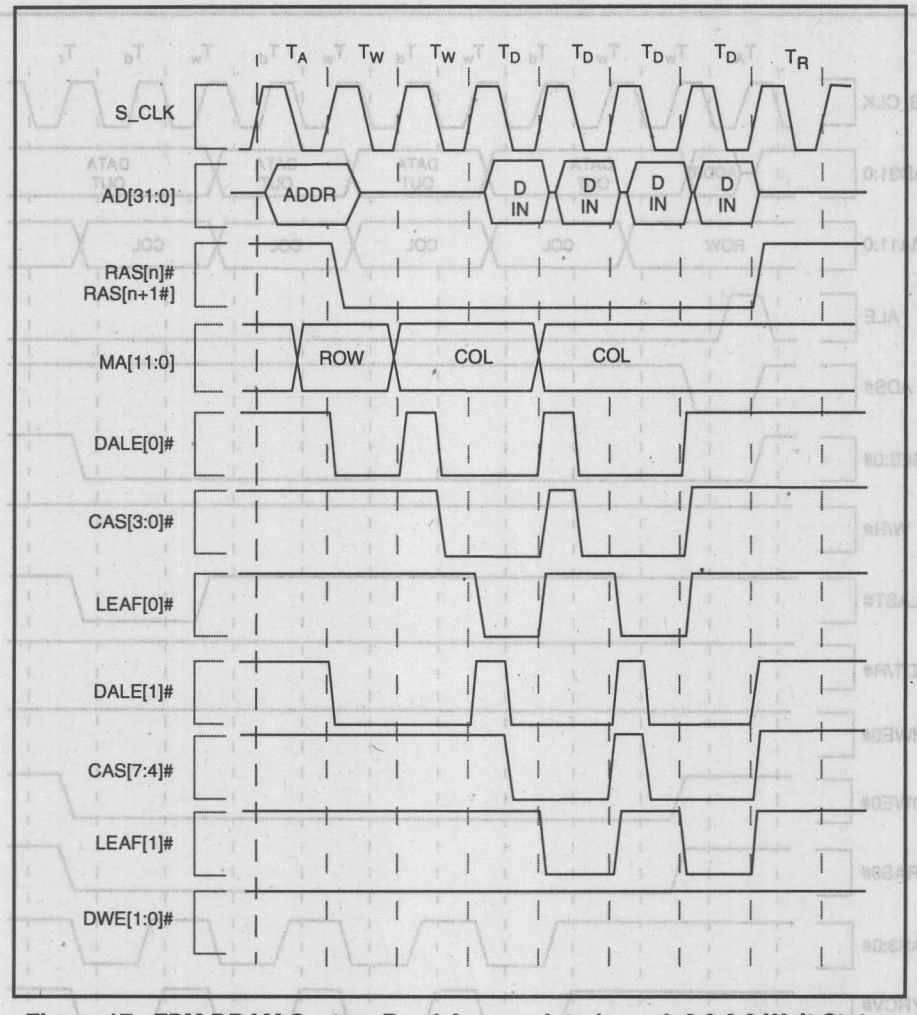


Figure 17. FPM DRAM System Read Access, Interleaved, 2,0,0,0 Wait States

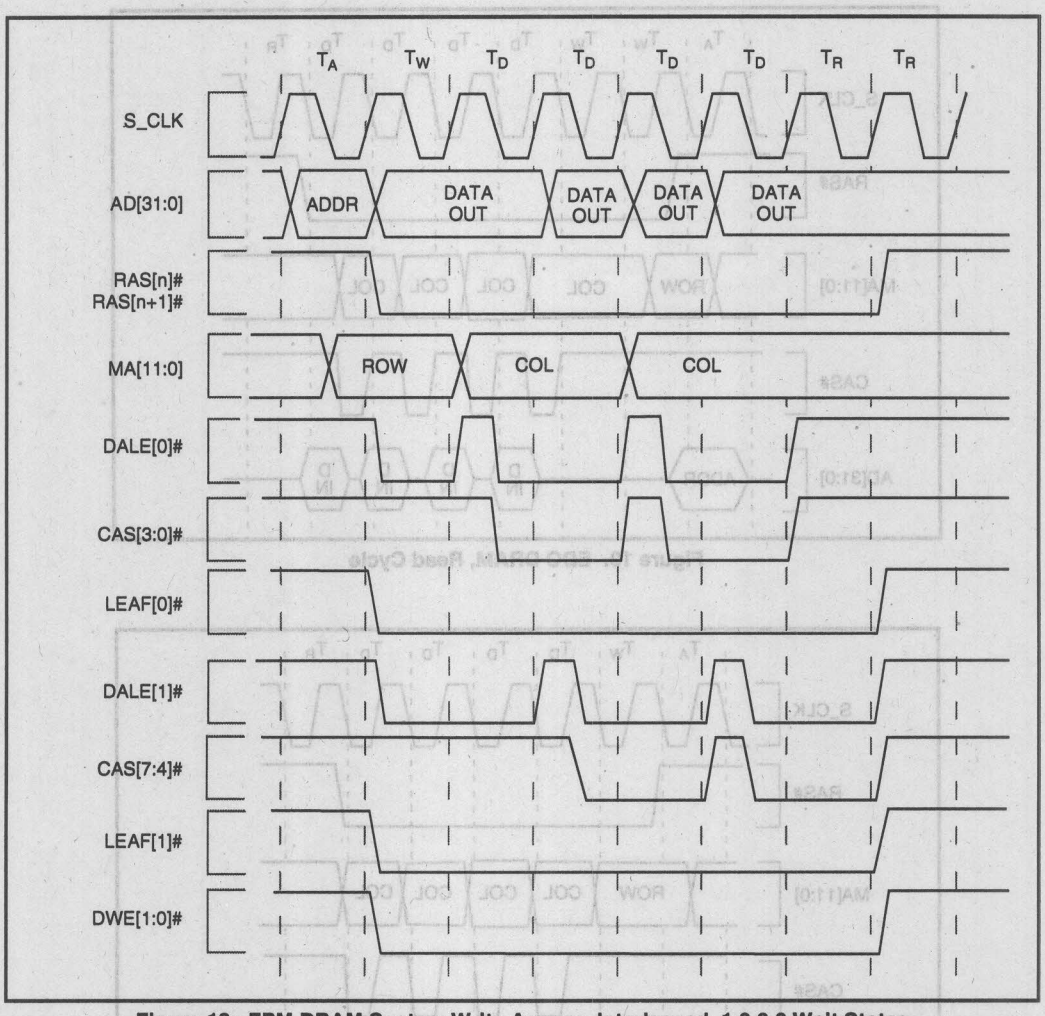


Figure 18. FPM DRAM System Write Access, Interleaved, 1,0,0,0 Wait States

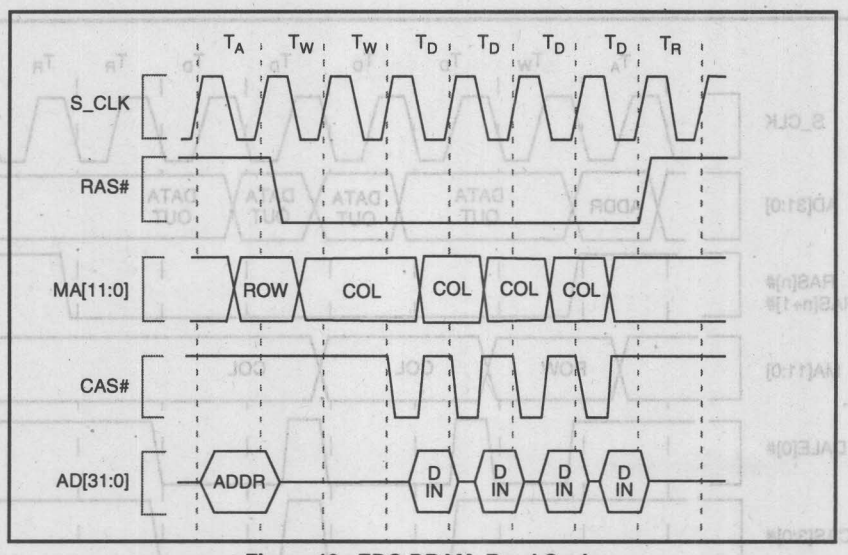


Figure 19. EDO DRAM, Read Cycle

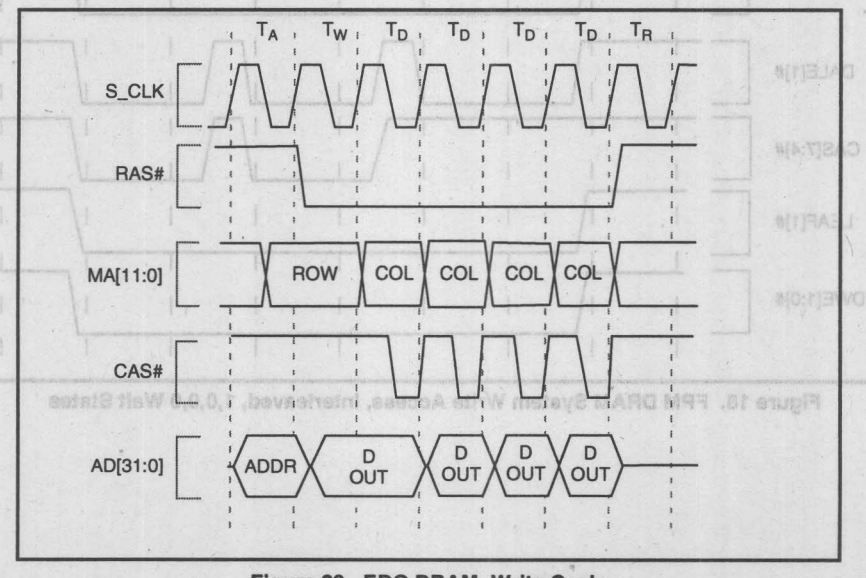


Figure 20. EDO DRAM, Write Cycle

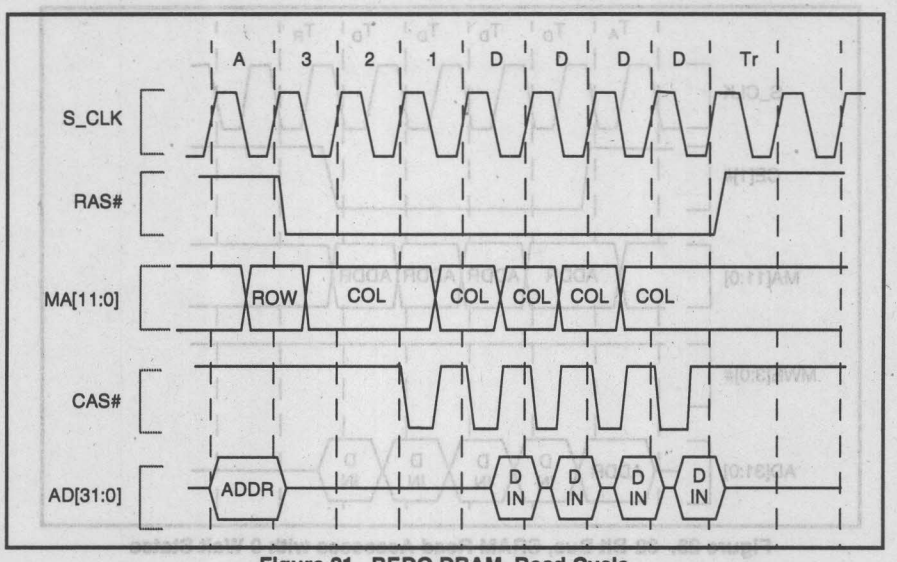


Figure 21. BEDO DRAM, Read Cycle

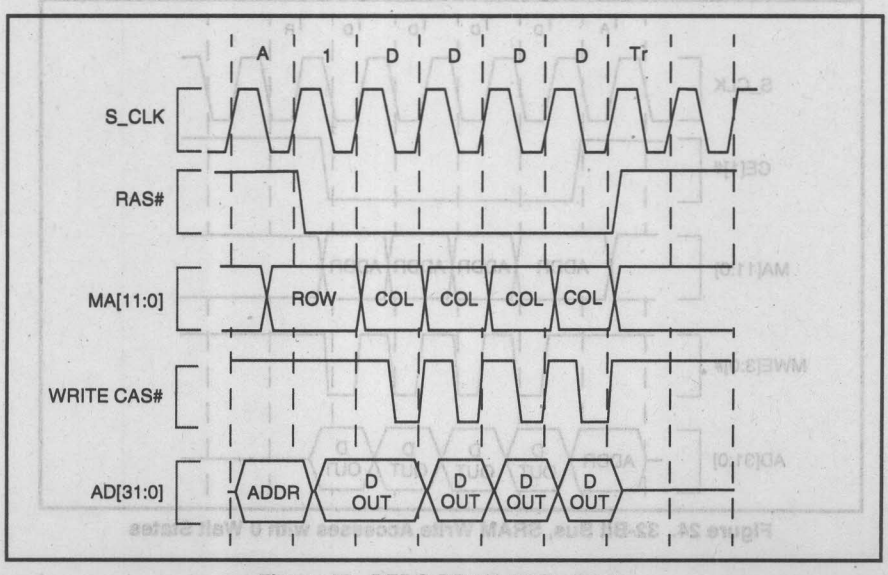


Figure 22. BEDO DRAM, Write Cycle

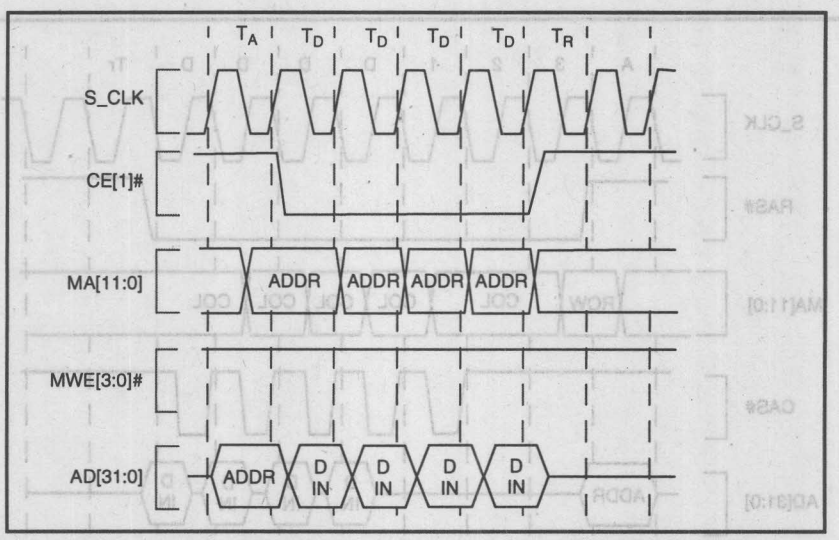


Figure 23. 32-Bit Bus, SRAM Read Accesses with 0 Wait States

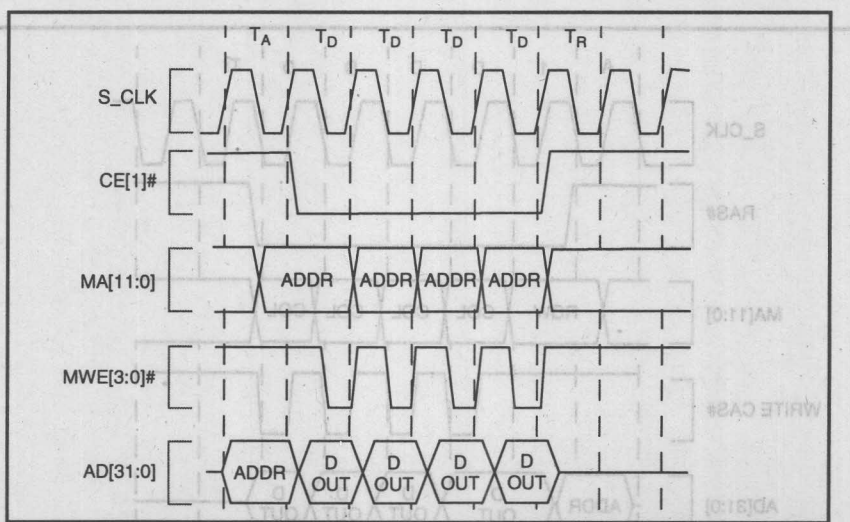


Figure 24. 32-Bit Bus, SRAM Write Accesses with 0 Wait States

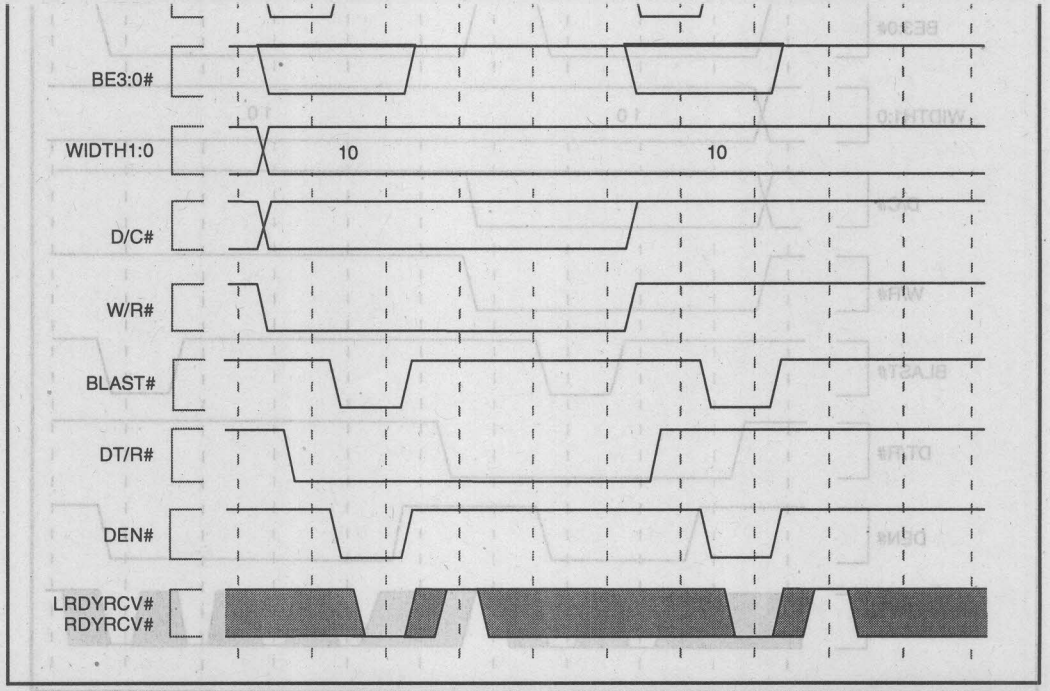


Figure 25. Non-Burst Read and Write Transactions without Wait States, 32-Bit 80960 Local Bus

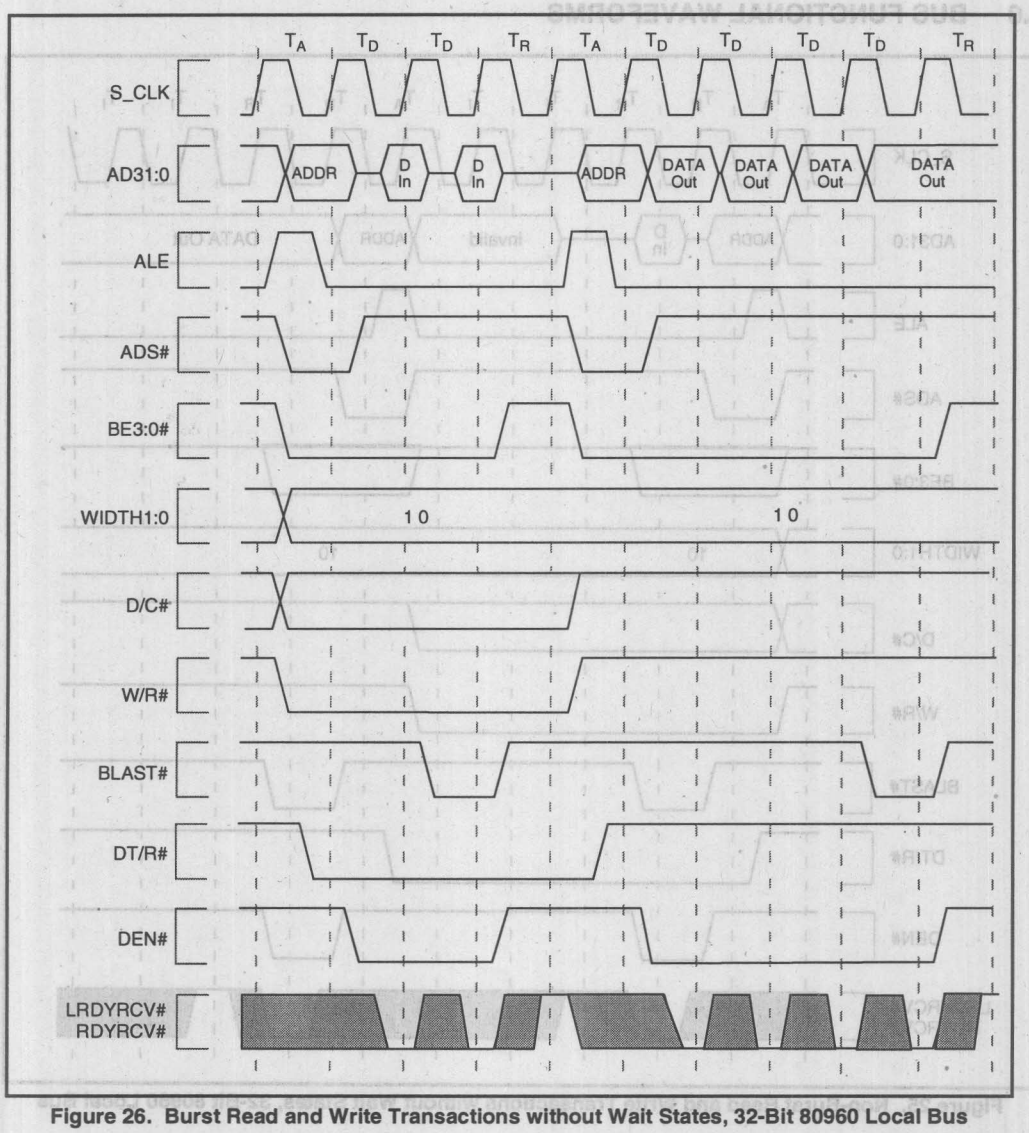


Figure 26. Burst Read and Write Transactions without Wait States, 32-Bit 80960 Local Bus

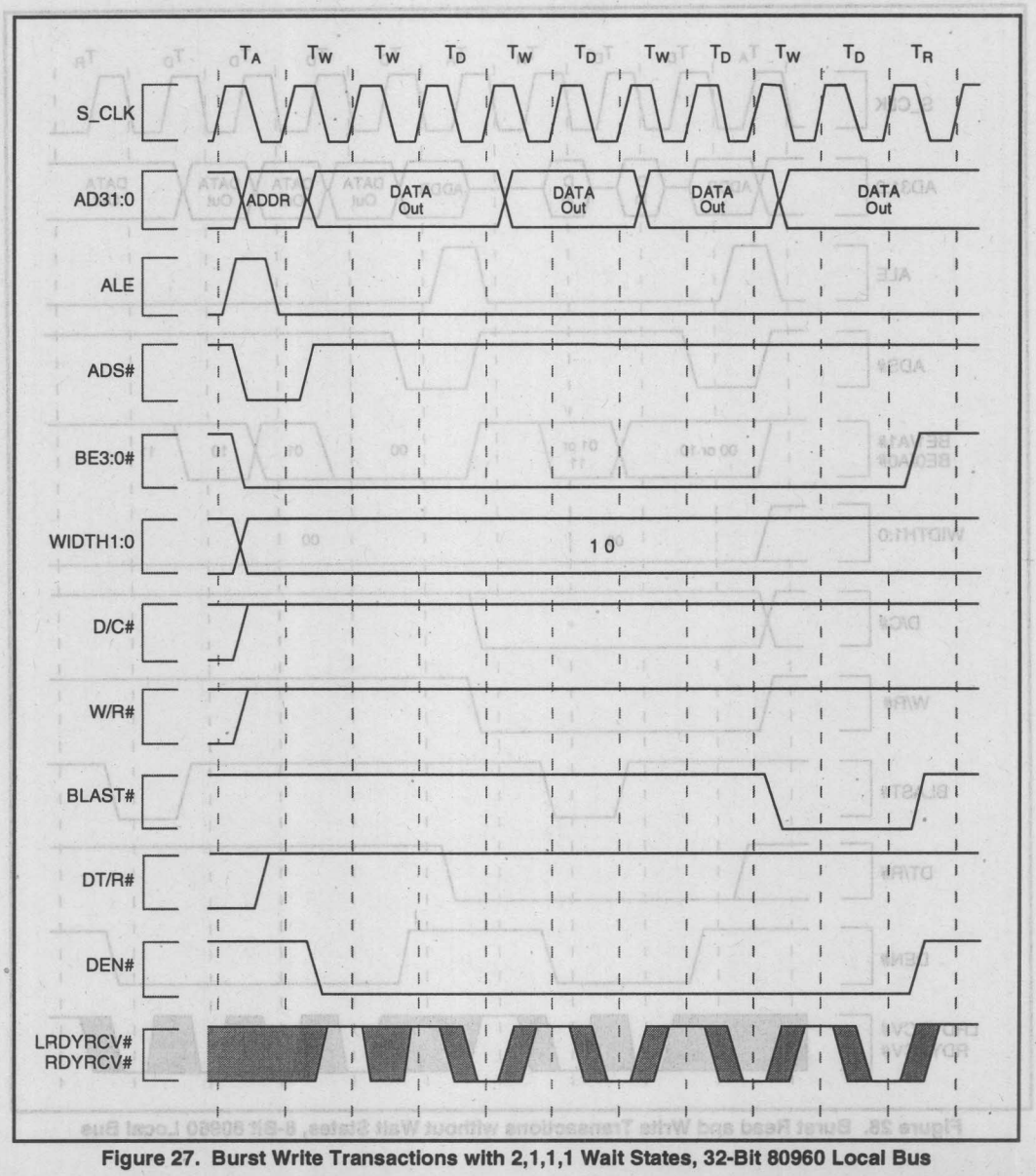


Figure 27. Burst Write Transactions with 2,1,1,1 Wait States, 32-Bit 80960 Local Bus

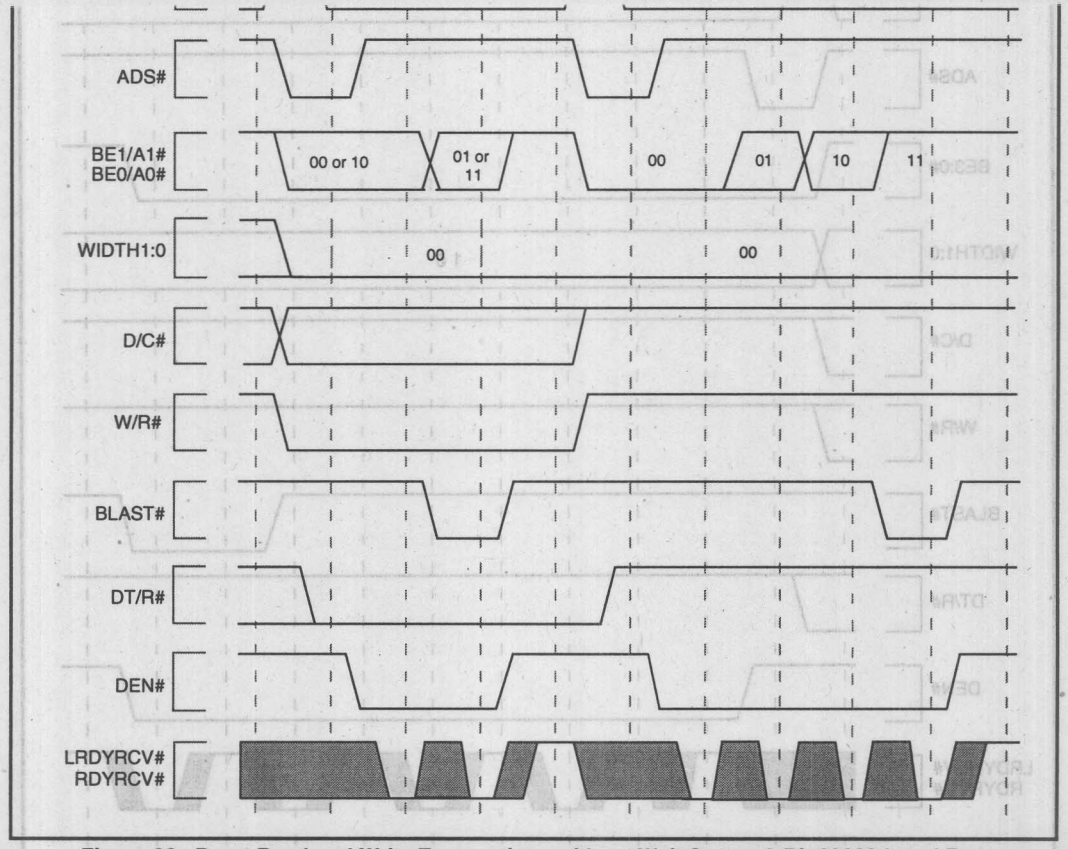


Figure 28. Burst Read and Write Transactions without Wait States, 8-Bit 80960 Local Bus

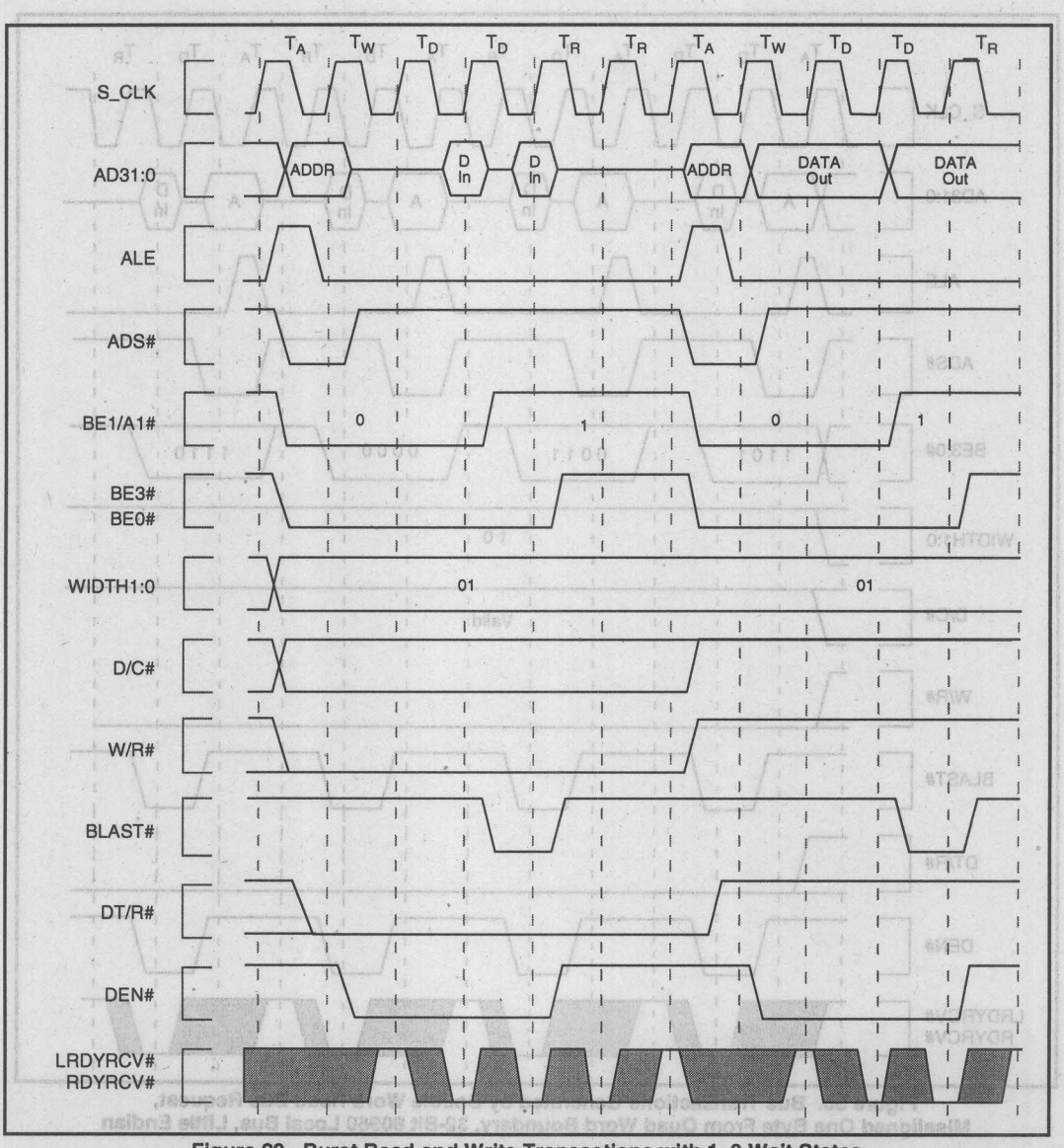


Figure 29. Burst Read and Write Transactions with 1, 0 Wait States and Extra Tr State on Read, 16-Bit 80960 Local Bus

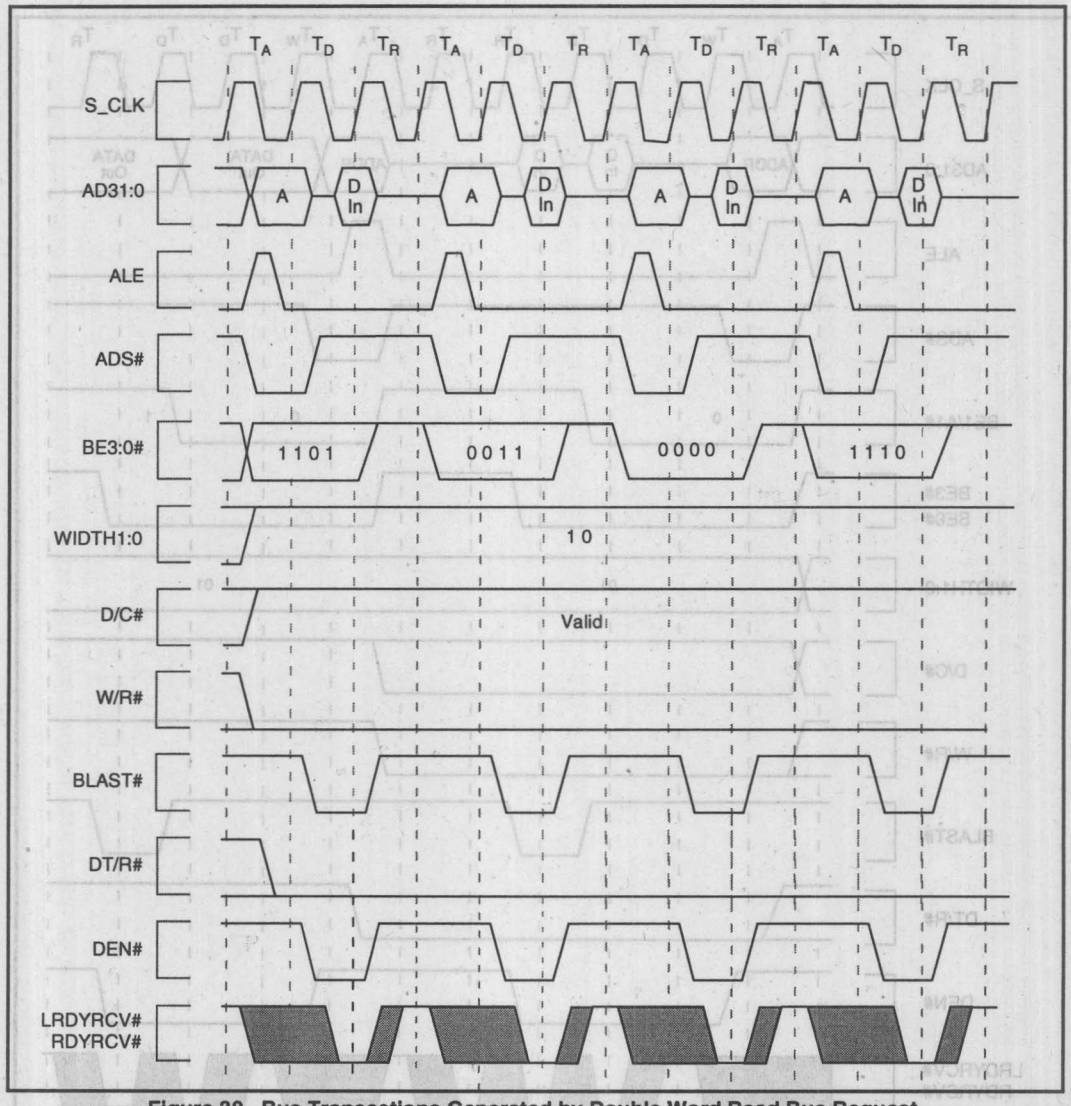
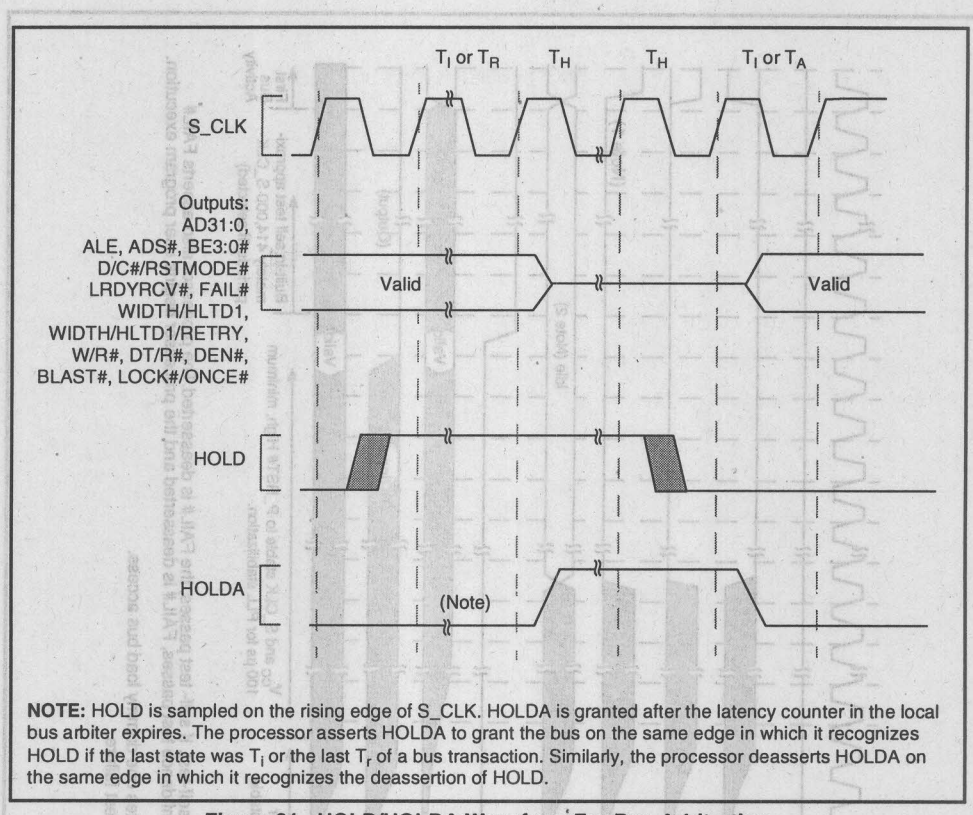
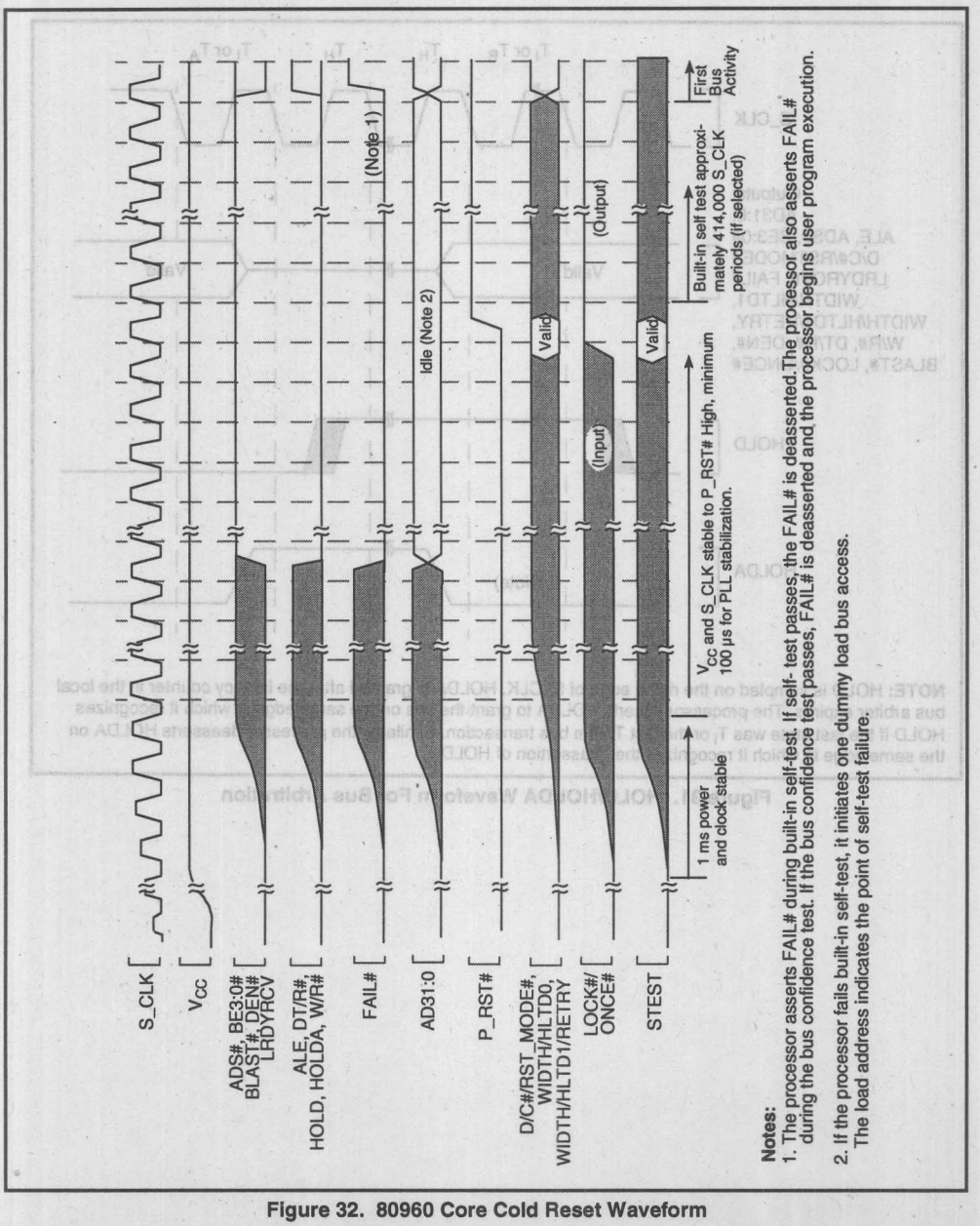


Figure 30. Bus Transactions Generated by Double Word Read Bus Request, Misaligned One Byte From Quad Word Boundary, 32-Bit 80960 Local Bus, Little Endian



NOTE: HOLD is sampled on the rising edge of S_CLK. HOLDA is granted after the latency counter in the local bus arbiter expires. The processor asserts HOLDA to grant the bus on the same edge in which it recognizes HOLD if the last state was T_i or the last T_r of a bus transaction. Similarly, the processor deasserts HOLDA on the same edge in which it recognizes the deassertion of HOLD.

Figure 31. HOLD/HOLDA Waveform For Bus Arbitration



Notes:
1. The processor asserts FAIL# during built-in self-test. If self- test passes, the FAIL# is deasserted. The processor also asserts FAIL# during the bus confidence test. If the bus confidence test passes, FAIL# is deasserted and the processor begins user program execution.
2. If the processor fails built-in self-test, it initiates one dummy load bus access. The load address indicates the point of self-test failure.

Figure 32. 80960 Core Cold Reset Waveform

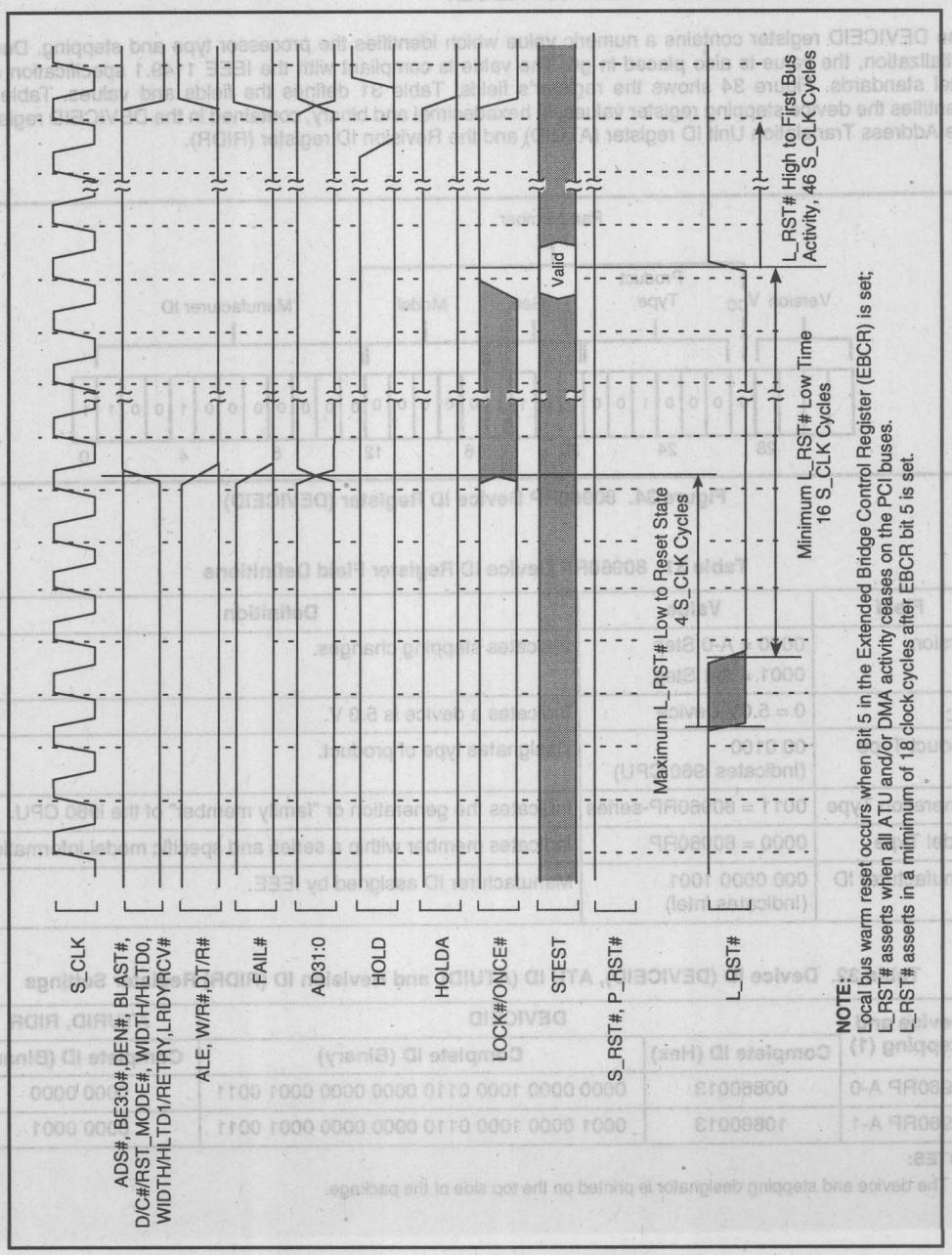


Figure 33. 80960 Local Bus Warm Reset Waveform

6.0 DEVICE IDENTIFICATION ON RESET

The DEVICEID register contains a numeric value which identifies the processor type and stepping. During initialization, the value is also placed in g0. The value is compliant with the IEEE 1149.1 specification and Intel standards. Figure 34 shows the register's fields; Table 31 defines the fields and values. Table 32 identifies the device stepping register values, in hexadecimal and binary, contained in the DEVICEID register, the Address Translation Unit ID register (ATUID) and the Revision ID register (RIDR).

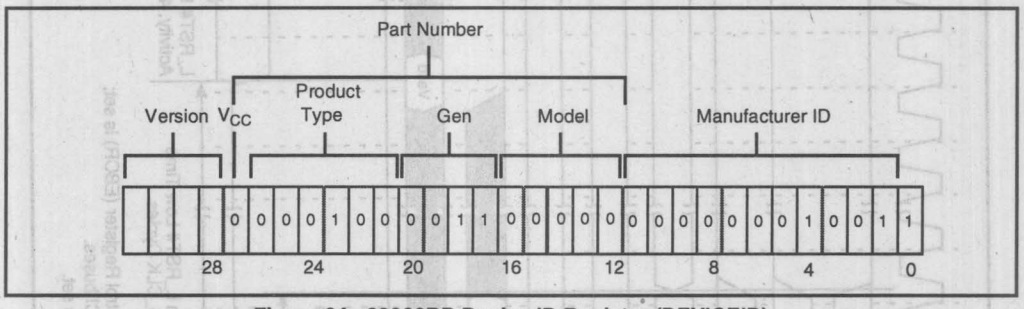


Figure 34. 80960RP Device ID Register (DEVICEID)

Table 31. 80960RP Device ID Register Field Definitions

Field	Value	Definition
Version	0000 = A-0 Step 0001 = A-1 Step	Indicates stepping changes.
V_{CC}	0 = 5.0V Device	Indicates a device is 5.0 V.
Product Type	00 0100 (Indicates i960 CPU)	Designates type of product.
Generation Type	0011 = 80960RP-series	Indicates the generation or "family member" of the i960 CPU.
Model Type	0000 = 80960RP	Indicates member within a series and specific model information.
Manufacturer ID	000 0000 1001 (Indicates Intel)	Manufacturer ID assigned by IEEE.

Table 32. Device ID (DEVICEID), ATU ID (ATUID), and Revision ID (RIDR) Register Settings

Device and Stepping (1)	DEVICEID		ATURID, RIDR
	Complete ID (Hex)	Complete ID (Binary)	Complete ID (Binary)
80960RP A-0	00860013	0000 0000 1000 0110 0000 0000 0001 0011	0000 0000
80960RP A-1	10860013	0001 0000 1000 0110 0000 0000 0001 0011	0000 0001

NOTES:
1. The device and stepping designator is printed on the top side of the package.

7.0 REVISION HISTORY

Table 33. Revision History (Rev. -001 to -002)

Section	Description
3.1.1, Functional Signal Definitions (pg. 8)	In all signal description tables, signals are now listed alphabetically and general clarification changes have been made.
Table 5., Power Requirement, Processor Control and Test Signal Descriptions (pg. 13)	Added new entries for L_RST# and $V_{CCPLL3:1}$.
Table 10., Clock Signal (pg. 20)	Created new table and redefined S_CLK clock signal.
Table 11., ICE Signal Descriptions (pg. 20)	Added table to define ICE signals.
3.1.2, 352-Lead HL-PBGA Package (pg. 21)	Modified section to include new package illustrations; added signal-to-ball cross reference tables.
3.2, Package Thermal Specifications (pg. 31)	Added thermal specifications section.
4.0, ELECTRICAL SPECIFICATIONS (pg. 34)	Added DC, AC specifications and related timing diagrams. These specifications are derived from simulations.
5.0, BUS FUNCTIONAL WAVEFORMS (pg. 55)	Removed several PCI-related waveforms.
Figure 33., 80960 Local Bus Warm Reset Waveform (pg. 63)	Added waveform.
6.0, DEVICE IDENTIFICATION ON RESET (pg. 64)	Added section.

Table 34. Revision History (Rev. -002 to -003) (Sheet 1 of 2)

Section	Description
(throughout document)	Removed all references to P_CLK and SYNC.
Table 4, Signal Descriptions (pg. 9)	Revised signal Descriptions for BE3:0#, DEN#, D/C#/RST_MODE#,LOCK#/ONCE#, LRDYRCV#, WIDTH/HLTD0 and WIDTH/HLTD1/RETRY
Table 5, Power Requirement, Processor Control and Test Signal Descriptions (pg. 13)	Revised signal Descriptions for TDI, TDO and TMS
Table 7, PCI Signal Descriptions (pg. 15)	Added new Type entries for P_LOCK#. Removed Type entries K(Q) or K(Z) from several signals. Changed P_CLK to S_CLK in P_INT[A:D]# and P_RST#. Revised Description for S_RST#. Changed S_RST# to P_RST# in S_REQ5#/ S_ARB_EN.
Table 8, Memory Controller Signal Descriptions (pg. 18)	Revised Descriptions for CAS7:0# and RAS3:0#.

Table 34. Revision History (Rev. -002 to -003) (Sheet 2 of 2)

Section	Description
3.1.2, 352-Lead HL-PBGA Package (pg. 21)	Updated section to show new package type, including new figures and tables. Specific signal name changes are: PCLK is renamed to VCC or VSS; WIDTH/HLTD0/SYNC is renamed to WIDTH/HLTD0
Section 3.2, Package Thermal Specifications (pg. 31)	Revised text and added figures and table.
Section 4.1, Absolute Maximum Ratings (pg. 34)	Revised Case Temperature from 85° C to 115° C.
Table 16, Operating Conditions (pg. 34)	Revised T_C Max.
Table 17, DC Characteristics (pg. 35)	Set values for V_{OL4}, I_{OL4} and V_{OH4}, I_{OH4} TBDs.
Table 18, I_{CC} Characteristics (pg. 36)	Set values for parameters which were previously TBDs.
Table 19, Input Clock Timings (pg. 37)	Removed T_F note.
Table 20, Synchronous Output Timings (pg. 37)	Revised T_{OV3} Max.
Table 21, Synchronous Input Timings (pg. 38)	Revised Min for T_{IS1} and T_{IS6}. Added Note 5 to T_{IS1}, T_{IS6} and T_{IH6}. Revised parameters for T_{IS3} and T_{IH3}. Added T_{IS9} and T_{IH9}.
Table 23, Fast Page Mode Non-interleaved DRAM Output Timings (pg. 39)	Revised all Min values. Revised all Max values except T_{OV9} and T_{OV11}.
Table 24, Fast Page Mode Interleaved DRAM Output Timings (pg. 39)	Revised all Min values. Revised all Max values for T_{OV12}, T_{OV13}, T_{OV14} and T_{OV16}.
Table 25, EDO DRAM Output Timings (pg. 40)	Revised all Min values. Revised all Max values except $T_{OV}27$ and $T_{OV}30$.
Table 26, BEDO DRAM Output Timings (pg. 41)	Revised all Min values.
Table 27, SRAM/ROM Output Timings (pg. 41)	Revised all Min values.
Table 28, Boundary Scan Test Signal Timings (pg. 42)	Revised Min values for T_{BSF}, T_{BSIS1}, T_{BSIH1}, T_{BSIS2} and T_{BSIH2}. Revised Max values for T_{BSOF1} and T_{BSOF2}.
Section 4.6, Memory Controller Output Timing Waveforms (pg. 48)	Revised Figure 15, (pg. 48) to Figure 24, (pg. 54).
Section 5.0, BUS FUNCTIONAL WAVEFORMS (pg. 55)	Revised Figure 31, (pg. 61) to Figure 33, (pg. 63)
Section 6.0, DEVICE IDENTIFICATION ON RESET (pg. 64)	Added Figure 34, (pg. 64) Revised Table 31, 80960RP Device ID Register Field Definitions (pg. 64).

ADVANCE INFORMATION

80960HA/HD/HT
32-BIT HIGH-PERFORMANCE SUPERSCALAR PROCESSOR

- 32-Bit Parallel Architecture
  - Load/Store Architecture
  - Sixteen 32-Bit Global Registers
  - Sixteen 32-Bit Local Registers
  - 1.2 Gbyte Internal Bandwidth (75 MHz)
  - On-Chip Register Cache
- Processor Core Clock
  - 80960HA is 1x Bus Clock
  - 80960HD is 2x Bus Clock
  - 80960HT is 3x Bus Clock
- Binary Compatible with Other 80960 Processors
- Issue Up To 150 Million Instructions per Second
- High-Performance On-Chip Storage
  - 16 Kbyte Four-Way Set-Associative Instruction Cache
  - 8 Kbyte Four-Way Set-Associative Data Cache
  - 2 Kbyte General Purpose RAM
  - Separate 128-Bit Internal Paths For Instructions/Data
- 3.3 V Supply Voltage
  - 5 V Tolerant Inputs
  - TTL Compatible Outputs
- Guarded Memory Unit
  - Provides Memory Protection
  - User/Supervisor Read/Write/Execute
- 32-Bit Demultiplexed Burst Bus
  - Per-Byte Parity Generation/Checking
  - Address Pipelining Option
  - Fully Programmable Wait State Generator
  - Supports 8-, 16- or 32-Bit Bus Widths
  - 160 Mbyte/s External Bandwidth (40 MHz)
- High-Speed Interrupt Controller
  - Up to 240 External Interrupts
  - 31 Fully Programmable Priorities
  - Separate, Non-maskable Interrupt Pin
- Dual On-Chip 32-Bit Timers
  - Auto Reload Capability and One-Shot
  - CLKIN Prescaling, ÷1, 2, 4 or 8
- JTAG Support - IEEE 1149.1 Compliant

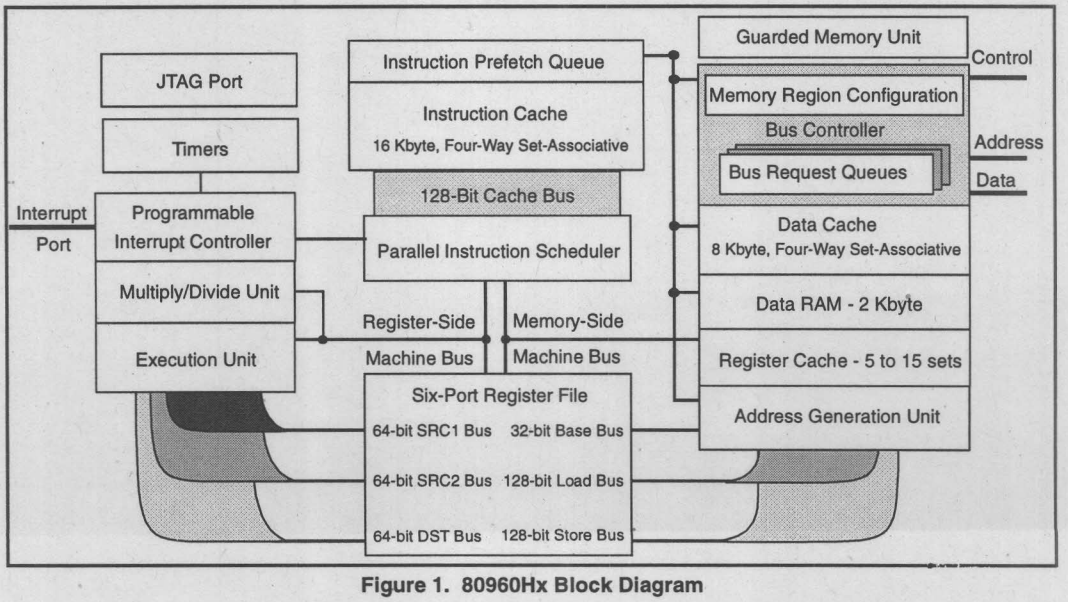


Figure 1. 80960Hx Block Diagram

© INTEL CORPORATION, 1996

June 1996

Order Number: 272495-004

80960HA/HD/HT

32-BIT HIGH-PERFORMANCE SUPERSCALAR PROCESSOR

- 32-Bit Parallel Architecture
  - Load/Store Architecture
  - Sixteen 32-Bit Global Registers
  - Sixteen 32-Bit Local Registers
  - 1.2 Gbyte Internal Bandwidth (75 MHz)
  - On-Chip Register Cache
- Processor Core Clock
  - 80960HA is 1x Bus Clock
  - 80960HD is 2x Bus Clock
  - 80960HT is 3x Bus Clock
- Binary Compatible with Other 80960 Processors
- Issue Up To 150 Million Instructions per Second
- High-Performance On-Chip Storage
  - 16 Kbyte Four-Way Set-Associative Instruction Cache
  - 8-Kbyte Four-Way Set-Associative Data Cache
  - 2 Kbyte General Purpose RAM
  - Separate 128-Bit Internal Paths For Instructions/Data
- 3.3 V Supply Voltage
  - 5 V Tolerant Inputs
  - TTL Compatible Outputs
- Guarded Memory Unit
  - Provides Memory Protection
  - User/Supervisor Read/Write/Execute
- 32-Bit Demultiplexed Burst Bus
  - Par-Byte Parity Generation/Checking
  - Address Pipelining Option
  - Fully Programmable Wait State Generator
  - Supports 8-, 16- or 32-Bit Bus Widths
  - 160 Mbyte/s External Bandwidth (40 MHz)
- High-Speed Interrupt Controller
  - Up to 240 External Interrupts
  - 31 Fully Programmable Priorities
  - Separate, Non-maskable Interrupt Pin
- Dual On-Chip 32-Bit Timers
  - Auto Reload Capability and One-Shot
  - CLKIN Prescaling, ÷1, 2, 4 or 8
- JTAG Support - IEEE 1149.1 Compliant

Figure 1. 80960Hx Block Diagram

© INTEL CORPORATION, 1998

Order Number: 272483-004

1.0 ABOUT THIS DOCUMENT

This document provides a preview of Intel's 80960Hx embedded superscalar microprocessors. Future revisions of this document will provide targeted electrical characteristics. Detailed descriptions for functional topics — other than parametric performance — will be published in the *i960® Hx Microprocessor User's Guide* (272484).

In this document, "80960Hx" and "i960 Hx processor" refer to the products described in Table 1. Throughout this document, information that is specific to each is clearly indicated.

NOTE:

Intel retains the right to make changes to these specifications at any time. In particular, descriptions of features, timings, packaging, and pinouts do not imply a commitment to implement them. In fact, this specification does not imply a commitment by Intel to design, manufacture, or sell the product described herein.

2.0 INTEL'S 80960Hx PROCESSOR

Intel's 80960Hx processor provides new performance levels while maintaining backward compatibility (pin* and software) with the i960® CA/CF processors. This newest member of the family of i960 32-bit, RISC-style, embedded processors allows customers to create scalable designs that meet multiple price and performance points. This is accomplished by providing processors that can run at the bus speed or faster using Intel's clock multiplying technology (see Table 1).The 80960Hx core is capable of issuing 150 million instructions per second, using a sophisticated instruction scheduler that allows the processor to sustain a throughput of two instructions every core clock, with a peak performance of three instructions per clock. The 80960Hx-series comprises three processors, which differ in the ratio of core clock speed to external bus speed.

In addition to expanded clock frequency options, the 80960Hx provides essential enhancements for an emerging class of high-performance embedded applications. Features include a larger instruction cache, data cache, and data RAM than any other 80960 processor to date. It also boasts a 32-bit demultiplexed and pipelined burst bus, fast interrupt mechanism, guarded memory unit, wait state generator, dual programmable timers, ONCE and IEEE 1149.1-compliant boundary scan test and debug support, and new instructions.

* Though not drop-in replaceable. Customers can design systems that accept either 80960Hx or Cx processors.

Table 1. 80960Hx Product Description

Product	Core	Voltage	Operating Frequency (bus/core)
80960HA	1x	3.3 V*	25/25, 33/33, 40/40
80960HD	2x	3.3 V*	16/32, 25/50, 33/66
80960HT	3x	3.3 V*	20/60, 25/75

NOTES: *Processor inputs are 5 V tolerant.

2.1 The i960 Processor Family

The i960 processor family is a 32-bit RISC architecture created by Intel to serve the needs of embedded applications. The embedded market includes applications as diverse as industrial automation, avionics, image processing, graphics and communications.

Because all members of the i960 processor family share a common core architecture, i960 applications are code-compatible. Each new processor in the family adds its own special set of functions to the core to satisfy the needs of a specific application or range of applications in the embedded market.

2.2 Key 80960Hx Features

2.2.1 Execution Architecture

Independent instruction paths inside the processor allow the execution of multiple, out-of-sequence instructions per clock. Register and resource scoreboarding interlocks maintain the logical integrity of sequential instructions that are being executed in parallel. To sustain execution of multiple instructions in each clock cycle, the processor decodes multiple instructions in parallel and simultaneously issues these instructions to parallel processing units. The

various processing units are then able to independently access instruction operands in parallel from a common register set.

Local Register Cache integrated on-chip provides automatic register management on call/return instructions. Upon a call instruction, the processor allocates a set of local registers for the called procedure, then stores the registers for the previous procedure in the on-chip register cache. As additional procedures are called, the cache stores the associated registers such that the most recently called procedure is the first available by the next return (**ret**) instruction. The processor can store up to fifteen register sets, after which the oldest sets are stored (spilled) into external memory.

The 80960Hx supports the 80960 architecturally-defined branch prediction mechanism. This allows many branches to execute with no pipeline break. With the 80960Hx's efficient pipeline, a branch can take as few as zero clocks to execute. The maximum penalty for an incorrect prediction is two core clocks.

2.2.2 Pipelined, Burst Bus

A 32-bit high performance bus controller interfaces the 80960Hx core to the external memory and peripherals. The Bus Control Unit features a maximum transfer rate of 160 Mbytes per second (at a 40 MHz external bus clock frequency). A key advantage of this design is its versatility. The user can independently program the physical and logical attributes of system memory. Physical attributes include wait state profile, bus width, and parity. Logical attributes include cacheability and Big or Little Endian byte order. Internally programmable wait states and 16 separately configurable physical memory regions allow the processor to interface with a variety of memory subsystems with minimum system complexity. To reduce the effect of wait states, the bus design is decoupled from the core. This lets the processor execute instructions while the bus performs memory accesses independently.

The Bus Controller's key features include:

- Demultiplexed, Burst Bus to support most efficient DRAM access modes
- Address Pipelining to reduce memory cost while maintaining performance
- 32-, 16- and 8-bit modes to facilitate I/O interfacing
- Full internal wait state generation to reduce system cost
- Little and Big Endian support
- Unaligned Access support implemented in hardware
- Three-deep request queue to decouple the bus from the core
- Independent physical and logical address space characteristics

2.2.3 On-Chip Caches and Data RAM

As shown in Figure 1, the 80960Hx provides generous on-chip cache and storage features to decouple CPU execution from the external bus. The processor includes a 16 Kbyte instruction cache, an 8 Kbyte data cache and 2 Kbytes of Data RAM. The caches are organized as 4-way set associative. Stores that hit the data cache are written through to memory. The data cache performs write allocation on cache misses. A fifteen-set stack frame cache allows the processor to rapidly allocate and deallocate local registers. All of the on-chip RAM sustains a 4-word (128-bit) access every clock cycle.

2.2.4 Priority Interrupt Controller

The interrupt unit provides the mechanism for the low latency and high throughput interrupt service essential for embedded applications. A priority interrupt controller provides full programmability of 240 interrupt sources with a typical interrupt task switch (latency) time of 17 core clocks. The controller supports 31 priority levels. Interrupts are prioritized and signaled within 10 core clocks of the request. If the interrupt has a higher priority than the processor priority, the context switch to the interrupt routine would typically complete in another 7 bus clocks.

External agents post interrupts via the 8-bit external interrupt port. The interrupt unit also handles the two internal sources from the Timers. Interrupts can be level- or edge-triggered.

2.2.5 Guarded Memory Unit

The Guarded Memory Unit (GMU) provides memory protection without the address translation found in Memory Management Units. The GMU contains two

memory protection schemes: one prevents illegal memory accesses, the other detects memory access violations. Both signal a fault to the processor. The programmable protection modes are: user read, write or execute; and supervisor read, write or execute.

2.2.6 Dual Programmable Timers

The processor provides two independent 32-bit timers, with four programmable clock rates. The user configures the timers via the Timer Unit registers. These registers are memory-mapped within the 80960Hx, addressable on 32-bit boundaries. The timers have a single-shot mode and auto-reload capabilities for continuous operation. Each timer has an independent interrupt request to the processor's interrupt controller.

2.2.7 Processor Self Test

When a system error is detected, the $\overline{FAIL}$ pin is asserted, a fail code message is driven onto the address bus, and the processor stops execution at the point of failure. The only way to resume normal operation is to perform a RESET operation. Because System Error generation can occur sometime after the bus confidence test and even after initialization during normal processor operation, the $\overline{FAIL}$ pin will be HIGH (logic "1") before the detection of a System Error.

The processor uses only one read bus-transaction to signal the fail code message; the address of the bus transaction is the fail code itself. The fail code is of the form: **0xfeffff***nn*; bits 6 to 0 contain a mask recording the possible failures. Bit 7, when set to 1, indicates that the mask contains failures from the internal Built-In Self-Test (BIST); when 0, the mask indicates other failures.

Systems should ignore reserved bits 0 and 1. This is also true for bits 5 and 6 when bit 7 is clear (=0).

The mask is shown in Table 2 and Table 3.

Table 2. Fail Codes For BIST (bit 7 = 1)

Bit	When Set:
6	On-chip Data-RAM failure detected by BIST.
5	Internal Microcode ROM failure detected by BIST.
4	Instruction cache failure detected by BIST.
3	Data cache failure detected by BIST.
2	Local-register cache or processor core failure detected by BIST.
1	Reserved. Always zero.
0	Reserved. Always zero.

Table 3. Remaining Fail Codes (bit 7 = 0)

Bit	When Set:
6	Reserved. Always one.
5	Reserved. Always one.
4	A data structure within the IMI is not aligned to a word boundary.
3	A System Error during normal operation has occurred.
2	The Bus Confidence test has failed.
1	Reserved. Always zero.
0	Reserved. Always zero.

2.3 Instruction Set Summary

Table 4 summarizes the 80960Hx instruction set by logical groupings.

Table 4. 80960Hx Instruction Set

Data Movement	Arithmetic	Logical	Bit / Bit Field / Byte
Load	Add	And	Set Bit
Store	Subtract	Not And	Clear Bit
Move	Multiply	And Not	Not Bit
Load Address	Divide	Or	Alter Bit
Conditional Select[2]	Remainder	Exclusive Or	Scan For Bit
	Modulo	Not Or	Span Over Bit
	Shift	Or Not	Extract
	Extended Shift	Nor	Modify
	Extended Multiply	Exclusive Nor	Scan Byte for Equal
	Extended Divide	Not	Byte Swap[2]
	Add with Carry	Nand	
	Subtract with Carry		
	Rotate		
	Conditional Add[2]		
	Conditional Subtract[2]		
Comparison	**Branch**	**Call/Return**	**Fault**
Compare	Unconditional Branch	Call	Conditional Fault
Conditional Compare	Conditional Branch	Call Extended	Synchronize Faults
Compare and Increment	Compare and Branch	Call System	
Compare and Decrement		Return	
Compare Byte[2]		Branch and Link	
Compare Short[2]			
Test Condition Code			
Check Bit			
Debug	**Processor Mgmt**	**Atomic**	**Cache Control**
Modify Trace Controls	Flush Local Registers	Atomic Add	Instruction Cache Control[1,2]
Mark	Modify Arithmetic Controls	Atomic Modify	Data Cache Control[1,2]
Force Mark	Modify Process Controls		
	Interrupt Enable/ Disable[1,2]		
	System Control[1]		

NOTES:
1) 80960Hx extensions to the 80960 core instruction set.
2) 80960Hx extensions to the 80960Cx instruction set.

3.0 PACKAGE INFORMATION

This section describes the pins, pinouts and thermal characteristics for the 80960Hx in the 168-pin ceramic Pin Grid Array (PGA) package, 208-pin PowerQuad2* (PQ2). For complete package specifications and information, see the Intel *Packaging* Handbook (Order# 240800).

3.1 Pin Descriptions

This section defines the 80960Hx pins. Table 5 presents the legend for interpreting the pin descriptions in Table 6. All pins float while the processor is in the ONCE mode, except TDO, which can be driven active according to normal JTAG specifications.

Table 5. Pin Description Nomenclature

Symbol	Description
I	Input only pin.
O	Output only pin.
I/O	Pin can be input or output.
-	Pin must be connected as indicated for proper device functionality.
S(E)	Synchronous edge sensitive input. This input must meet the setup and hold times relative to CLKIN to ensure proper operation of the processor.
S(L)	Synchronous level sensitive input. This input must meet the setup and hold times relative to CLKIN to ensure proper operation of the processor.
A(E)	Asynchronous edge-sensitive input.
A(L)	Asynchronous level-sensitive input.
H(...)	While the processor bus is in the HOLD state (HLDA asserted), the pin: H(1) is driven to V_{CC} H(0) is driven to V_{SS} H(Z) floats H(Q) continues to be a valid output
B(...)	While the processor is in the bus backoff state ($\overline{BOFF}$ asserted), the pin: B(1) is driven to V_{CC} B(0) is driven to V_{SS} B(Z) floats B(Q) continues to be a valid output
R(...)	While the processor's $\overline{RESET}$ pin is asserted, the pin: R(1) is driven to V_{CC} R(0) is driven to V_{SS} R(Z) floats R(Q) continues to be a valid output

*PowerQuad is a trademark of Amkor Electronics.

Table 6. 80960Hx Processor Family Pin Descriptions (Sheet 1 of 5)

Name	Type	Description
A31:2	O H(Z) B(Z) R(Z)	**ADDRESS BUS** carries the upper 30 bits of the physical address. A31 is the most significant address bit and A2 is the least significant. During a bus access, A31:2 identify all external addresses to word (4-byte) boundaries. The byte enable signals indicate the selected byte in each word. During burst accesses, A3 and A2 increment to indicate successive addresses.
D31:0	I/O S(L) H(Z) B(Z) R(Z)	**DATA BUS** carries 32, 16, or 8-bit data quantities depending on bus width configuration. The least significant bit of the data is carried on D0 and the most significant on D31. The lower 8 data lines (D7:0) are used when the bus is configured for 8-bit data. When configured for 16-bit data, D15:0 are used.
DP3:0	I/O S(L) H(Z) B(Z) R(Z)	**DATA PARITY** carries parity information for the data bus. Each parity bit is assigned a group of 8 data bus pins as follows: DP3 generates/checks parity for D31:24 DP2 generates/checks parity for D23:16 DP1 generates/checks parity for D15:8 DP0 generates/checks parity for D7:0 Parity information is generated for a processor write cycle and is checked for a processor read cycle. Parity checking and polarity are programmable. Parity generation/checking is only performed for the size of the data accessed.
$\overline{\text{PCHK}}$	O H(Q) B(Q) R(1)	**PARITY CHECK** indicates the result of a parity check operation. An asserted $\overline{\text{PCHK}}$ indicates that the previous bus read access resulted in a parity check error.
$\overline{\text{BE3:0}}$	O H(Z) B(Z) R(1)	**BYTE ENABLES** select which of the four bytes addressed by A31:2 are active during a bus access. Byte enable encoding is dependent on the bus width of the memory region accessed: *32-bit bus:* $\overline{\text{BE3}}$ enables D31:24 $\overline{\text{BE2}}$ enables D23:16 $\overline{\text{BE1}}$ enables D15:8 $\overline{\text{BE0}}$ enables D7:0 *16-bit bus:* $\overline{\text{BE3}}$ becomes Byte High Enable (enables D15:8) $\overline{\text{BE2}}$ is not used (state is undefined) $\overline{\text{BE1}}$ becomes Address Bit 1 (A1) $\overline{\text{BE0}}$ becomes Byte Low Enable (enables D7:0) *8-bit bus:* $\overline{\text{BE3}}$ is not used (state is undefined) $\overline{\text{BE2}}$ is not used (state is undefined) $\overline{\text{BE1}}$ Address Bit 1 (A1) $\overline{\text{BE0}}$ Address Bit 0 (A0)

Table 6. 80960Hx Processor Family Pin Descriptions (Sheet 2 of 5)

Name	Type	Description
W/$\overline{R}$	O H(Z) B(Z) R(0)	**WRITE/READ** is low for read accesses and high for write accesses. W/$\overline{R}$ becomes valid during the address phase of a bus cycle and remains valid until the end of the cycle for non-pipelined accesses. For pipelined accesses, W/$\overline{R}$ changes state when the next address is presented. 0= Read 1= Write
D/$\overline{C}$	O H(Z) B(Z) R(0)	**DATA/CODE** indicates that a bus access is a data access or an instruction access. D/$\overline{C}$ has the same timing as W/$\overline{R}$. 0 = Code 1 = Data
$\overline{SUP}$	O H(Z) B(Z) R(1)	**SUPERVISOR ACCESS** indicates whether the current bus access originates from a request issued while in supervisor mode or user mode. $\overline{SUP}$ can be used by the memory subsystem to isolate supervisor code and data structures from non-supervisor access. 0 = Supervisor Mode 1 = User Mode
$\overline{ADS}$	O H(Z) B(Z) R(1)	**ADDRESS STROBE** indicates a valid address and the start of a new bus access. $\overline{ADS}$ is asserted for the first clock of a bus access.
$\overline{READY}$	I S(L)	$\overline{READY}$, when enabled for a memory region, is asserted by the memory subsystem to indicate the completion of a data transfer. $\overline{READY}$ is used to indicate that read data on the bus is valid, or that a write transfer has completed. $\overline{READY}$ works in conjunction with the internal wait state generator to accommodate various memory speeds. $\overline{READY}$ is sampled after any programmed wait states: • During each data cycle of a burst access • During the data cycle of a non-burst access
$\overline{BTERM}$	I S(L)	**BURST TERMINATE**, when enabled for a memory region, is asserted by the memory subsystem to terminate a burst access in progress. When $\overline{BTERM}$ is asserted, the current burst access is terminated and another address cycle occurs.
$\overline{WAIT}$	O H(Z) B(Z) R(1)	$\overline{WAIT}$ indicates the status of the internal wait-state generator. $\overline{WAIT}$ is asserted when the internal wait state generator generates N_{WAD}, N_{RAD}, N_{WDD} and N_{RDD} wait states. $\overline{WAIT}$ can be used to derive a write data strobe.
$\overline{BLAST}$	O H(Z) B(Z) R(1)	**BURST LAST** indicates the last transfer in a bus access. $\overline{BLAST}$ is asserted in the last data transfer of burst and non-burst accesses after the internal wait-state generator reaches zero. $\overline{BLAST}$ remains active as long as wait states are inserted via the $\overline{READY}$ pin. $\overline{BLAST}$ becomes inactive after the final data transfer in a bus cycle.

Table 6. 80960Hx Processor Family Pin Descriptions (Sheet 3 of 5)

Name	Type	Description
DT/$\overline{R}$	O H(Z) B(Z) R(0)	**DATA TRANSMIT/RECEIVE** indicates direction for data transceivers. DT/$\overline{R}$ is used with $\overline{DEN}$ to provide control for data transceivers connected to the data bus. DT/$\overline{R}$ is driven low to indicate the processor expects data (a read cycle). DT/$\overline{R}$ is driven high when the processor is "transmitting" data (a store cycle). DT/$\overline{R}$ only changes state when $\overline{DEN}$ is high. 0 = Data Receive 1 = Data Transmit
$\overline{DEN}$	O H(Z) B(Z) R(1)	**DATA ENABLE** indicates data transfer cycles during a bus access. $\overline{DEN}$ is asserted at the start of the first data cycle in a bus access and de-asserted at the end of the last data cycle. $\overline{DEN}$ remains asserted for an entire bus request, even when that request spans several bus accesses. For example, a ldq instruction starting at an unaligned quad word boundary is one bus request spanning at least two bus accesses. $\overline{DEN}$ remains asserted throughout all the accesses (including $\overline{ADS}$ states) and de-asserts when the lqd instruction request is satisfied. $\overline{DEN}$ is used with DT/$\overline{R}$ to provide control for data transceivers connected to the data bus. $\overline{DEN}$ remains asserted for sequential reads from pipelined memory regions.
$\overline{LOCK}$	O H(Z) B(Z) R(1)	**BUS LOCK** indicates that an atomic read-modify-write operation is in progress. $\overline{LOCK}$ may be used by the memory subsystem to prevent external agents from accessing memory that is currently involved in an atomic operation (e.g., a semaphore). $\overline{LOCK}$ is asserted in the first clock of an atomic operation and de-asserted when $\overline{BLAST}$ is deasserted in the last bus cycle.
HOLD	I S(L)	**HOLD REQUEST** signals that an external agent requests access to the processor's address, data, and control buses. When HOLD is asserted, the processor: • Completes the current bus request. • Asserts HOLDA and floats the address, data, and control buses. When HOLD is deasserted, the HOLDA pin is deasserted and the processor reassumes control of the address, data, and control pins.
HOLDA	O H(1) B(0) R(Q)	**HOLD ACKNOWLEDGE** indicates to an external master that the processor has relinquished control of the bus. The processor grants HOLD requests and enters the HOLDA state while the $\overline{RESET}$ pin is asserted. HOLDA is never granted while $\overline{LOCK}$ is asserted.
$\overline{BOFF}$	I S(L)	**BUS BACKOFF** forces the processor to immediately relinquish control of the bus on the next clock cycle. When $\overline{READY}$/$\overline{BTERM}$ is enabled and: • When $\overline{BOFF}$ is asserted, the address, data, and control buses are floated on the next clock cycle and the current access is aborted. • When $\overline{BOFF}$ is deasserted, the processor resumes by regenerating the aborted bus access. See Figure 16, Bus Backoff ($\overline{BOFF}$) Timings (pg. 38), for $\overline{BOFF}$ timing requirements.
BREQ	O H(Q) B(Q) R(0)	**BUS REQUEST** indicates that a bus request is pending in the bus controller, but the processor *is not stalled* pending the result of the bus operation. BREQ can be used with BSTALL to indicate to an external bus arbiter the processor's bus ownership requirements.

Table 6. 80960Hx Processor Family Pin Descriptions (Sheet 4 of 5)

Name	Type	Description
BSTALL	O H(Q) B(Q) R(0)	**BUS STALL** indicates that the processor has stalled pending the result of a request in the bus controller. When BSTALL is asserted, the processor *must* regain bus ownership to continue processing (i.e., it can no longer execute strictly out of on-chip cache memory).
CT3:0	O H(Z) B(Z) R(Z) P(7)	**CYCLE TYPE** indicates the type of bus cycle currently being started or processor state. CT3:0 encoding follows: **Cycle Type** — **ADS** — **CT3:0** Program-initiated access using 8-bit bus — 0 — 0000 Program-initiated access using 16-bit bus — 0 — 0001 Program-initiated access using 32-bit bus — 0 — 0010 Event-initiated access using 8-bit bus — 0 — 0100 Event-initiated access using 16-bit bus — 0 — 0101 Event-initiated access using 32-bit bus — 0 — 0110 Reserved — 0 — 0X11 Reserved for future products — 0 — 1XXX Processor not halted, otherwise reserved — 1 — 0X00 Processor not halted, otherwise reserved — 1 — 0X01 Processor not halted, otherwise reserved — 1 — 0X10 Reserved — 1 — 0011 Reserved — 1 — 0111 Reserved for future products — 1 — 1XXX
$\overline{\text{XINT7:0}}$	I A(E) A(L)	**EXTERNAL INTERRUPT** pins are used to request interrupt service. These pins can be configured in three modes: *Dedicated Mode:* Each pin is assigned a dedicated interrupt level. Dedicated inputs can be programmed to be level (low or high) or edge (rising or falling) sensitive. *Expanded Mode:* All eight pins act as a vectored interrupt source. The interrupt pins are level sensitive in this mode. *Mixed Mode:* The $\overline{\text{XINT7:5}}$ pins act as dedicated sources and the $\overline{\text{XINT4:0}}$ pins act as the five most significant bits of a vectored source. The least significant bits of the vectored source are set to "010" internally.
$\overline{\text{NMI}}$	I A(E)	**NON-MASKABLE INTERRUPT** causes a non-maskable interrupt event to occur. $\overline{\text{NMI}}$ is the highest priority interrupt source. $\overline{\text{NMI}}$ is falling edge triggered.
CLKIN	I	**CLOCK INPUT** provides the time base for the 80960Hx. All internal circuitry is synchronized to CLKIN. All input and output timings are specified relative to CLKIN. For the 80960HD, the 2x internal clock is derived by multiplying the CLKIN frequency by 2. For the 80960HT, the 3x internal clock is derived by multiplying the CLKIN frequency by 3.
$\overline{\text{RESET}}$	I A(L)	**$\overline{\text{RESET}}$** forces the device into reset. $\overline{\text{RESET}}$ causes all external and internal signals to return to their reset state (if defined). The rising edge of $\overline{\text{RESET}}$ starts the processor boot sequence.
STEST	I S(L)	**SELF TEST**, when asserted during the rising edge of $\overline{\text{RESET}}$, causes the processor to execute its built in self-test.

	P(1)	is asserted and the processor ceases execution.
$\overline{\text{ONCE}}$	**I**	**ON-CIRCUIT EMULATION** control: the processor samples this pin during reset. If it is asserted low at the end of reset, the processor enters ONCE mode. In ONCE mode, the processor stops all clocks and floats all output pins except the TDO pin. $\overline{\text{ONCE}}$ includes a nominal 80 kΩ internal pull-up resistor.
TCK	**I**	**TEST CLOCK** provides the clocking function for IEEE 1149.1 Boundary Scan testing.
TDI	**I**	**TEST DATA INPUT** is the serial input pin for IEEE 1149.1 Boundary Scan testing.
TDO	**O**	**TEST DATA OUTPUT** is the serial output pin for IEEE 1149.1 Boundary Scan testing. ONCE does not disable this pin.
$\overline{\text{TRST}}$	**I**	**TEST RESET** asynchronously resets the Test Access Port (TAP) controller. $\overline{\text{TRST}}$ must be held low at least 10,000 clock cycles after power-up. One method is to provide $\overline{\text{TRST}}$ with a separate power-on-reset circuit. $\overline{\text{TRST}}$ includes a nominal 65 kΩ internal pull-up resistor. Pull this pin low when not in use.
TMS	**I**	**TEST MODE SELECT** is sampled at the rising edge of TCK. TCK controls the sequence of TAP controller state changes for IEEE 1149.1 Boundary Scan testing. TMS includes a nominal 80 kΩ internal pull-up resistor.
VCC5	**I**	**5 V REFERENCE VOLTAGE** input is the reference voltage for the 5 V-tolerant I/O buffers. This signal should be connected to +5 V for use with inputs which exceed 3.3 V. If all inputs are from 3.3 V components, this pin should be connected to 3.3 V.
VCCPLL	**I**	**PLL VOLTAGE** is the +3.3 VDC analog input for the PLL.
VOLDET	**O**	**VOLTAGE DETECT** signal allows external system logic to distinguish between a 5 V 80960Cx processor and the 3.3 V 80960Hx processor. This signal is active low for a 3.3 V 80960Hx (it is high impedance for 5 V 80960Cx). This pin is available only on the PGA version. 0 = 80960Hx 1 = 80960Cx

3.2 80960Hx Mechanical Data

3.2.1 80960Hx PGA Pinout

Figure 2 depicts the complete 80960Hx PGA pinout as viewed from the top side of the component (i.e., pins facing down). Figure 3 shows the complete 80960Hx PGA pinout as viewed from the pin-side of the package (i.e., pins facing up). Table 8 lists the 80960Hx pin names with package location. See Section 4.3, Recommended Connections (pg. 27) for specifications and recommended connections.

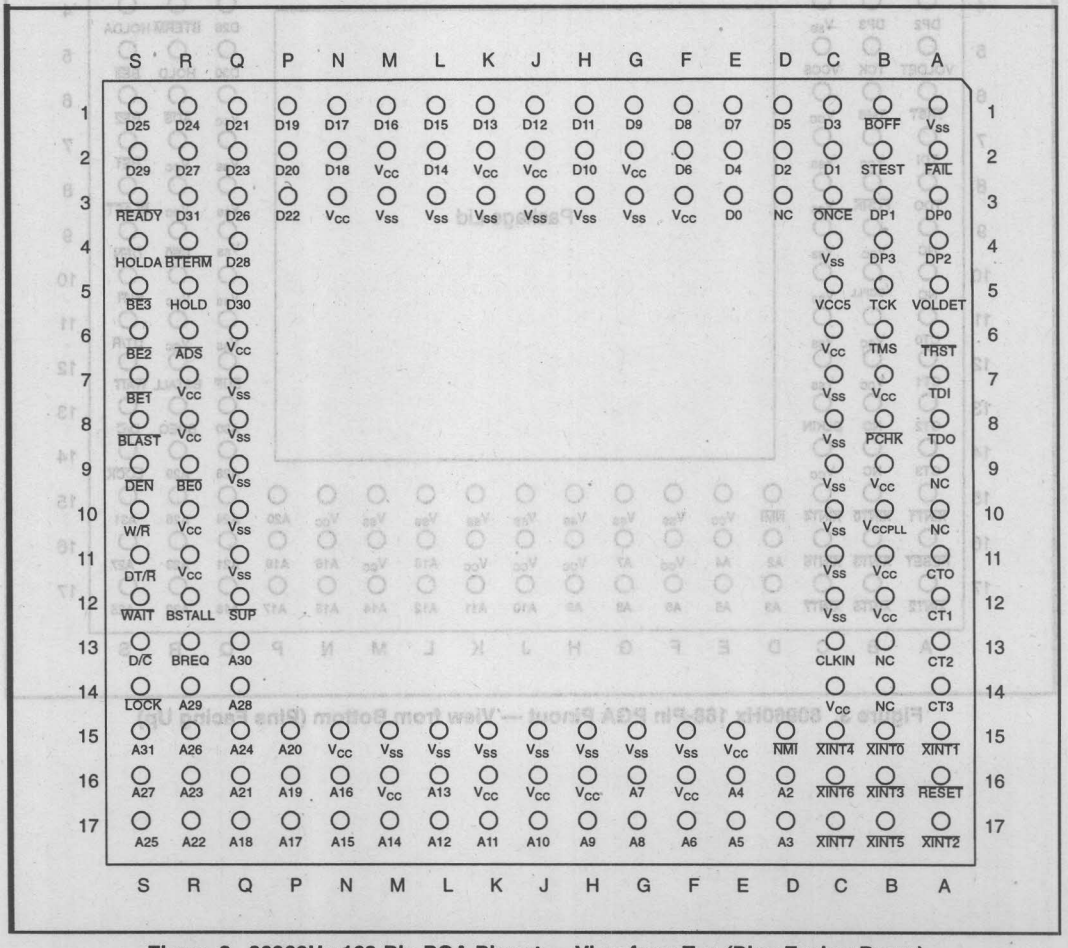


Figure 2. 80960Hx 168-Pin PGA Pinout — View from Top (Pins Facing Down)

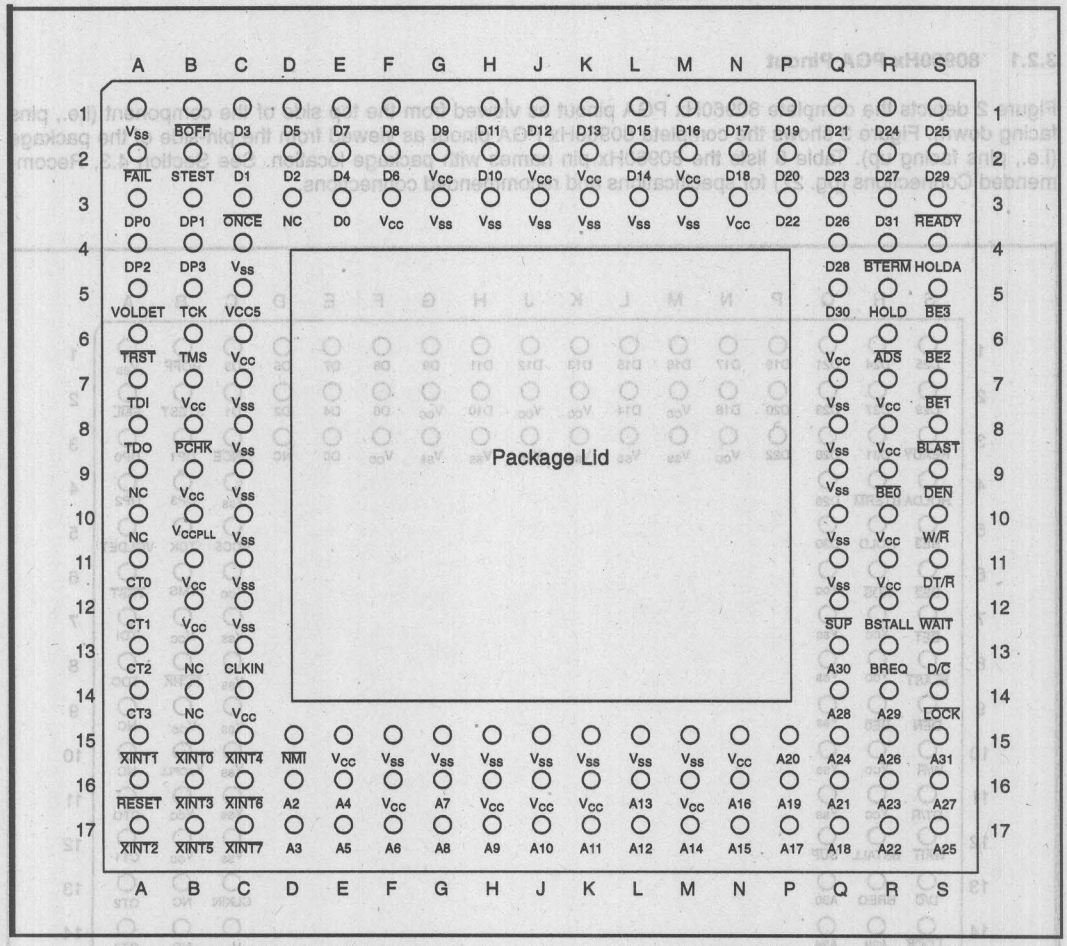


Figure 3. 80960Hx 168-Pin PGA Pinout — View from Bottom (Pins Facing Up)

Table 7. 80960Hx 168-Pin PGA Pinout — Signal Name Order (Sheet 1 of 2)

Signal Name	PGA Pin	Signal Name	PGA Pin	Signal Name	PGA Pin	Signal Name	PGA Pin
A2	D16	$\overline{ADS}$	R6	D14	L2	$\overline{LOCK}$	S14
A3	D17	$\overline{BE0}$	R9	D15	L1	NC	A9
A4	E16	$\overline{BE1}$	S7	D16	M1	NC	A10
A5	E17	$\overline{BE2}$	S6	D17	N1	NC	B13
A6	F17	$\overline{BE3}$	S5	D18	N2	NC	B14
A7	G16	$\overline{BLAST}$	S8	D19	P1	NC	D3
A8	G17	$\overline{BOFF}$	B1	D20	P2	$\overline{NMI}$	D15
A9	H17	BREQ	R13	D21	Q1	$\overline{ONCE}$	C3
A10	J17	BSTALL	R12	D22	P3	$\overline{PCHK}$	B8
A11	K17	$\overline{BTERM}$	R4	D23	Q2	$\overline{READY}$	S3
A12	L17	CLKIN	C13	D24	R1	$\overline{RESET}$	A16
A13	L16	CT0	A11	D25	S1	STEST	B2
A14	M17	CT1	A12	D26	Q3	$\overline{SUP}$	Q12
A15	N17	CT2	A13	D27	R2	TCK	B5
A16	N16	CT3	A14	D28	Q4	TDI	A7
A17	P17	D/$\overline{C}$	S13	D29	S2	TDO	A8
A18	Q17	D0	E3	D30	Q5	TMS	B6
A19	P16	D1	C2	D31	R3	$\overline{TRST}$	A6
A20	P15	D2	D2	$\overline{DEN}$	S9	V_{CC}	B7
A21	Q16	D3	C1	DP0	A3	V_{CC}	B9
A22	R17	D4	E2	DP1	B3	V_{CC}	B11
A23	R16	D5	D1	DP2	A4	V_{CC}	B12
A24	Q15	D6	F2	DP3	B4	V_{CC}	C6
A25	S17	D7	E1	DT/$\overline{R}$	S11	V_{CC}	C14
A26	R15	D8	F1	$\overline{FAIL}$	A2	V_{CC}	E15
A27	S16	D9	G1	--	--	V_{CC}	F3
A28	Q14	D10	H2	--	--	V_{CC}	F16
A29	R14	D11	H1	--	--	V_{CC}	G2
A30	Q13	D12	J1	HOLD	R5	V_{CC}	H16
A31	S15	D13	K1	HOLDA	S4	V_{CC}	J2

Table 7. 80960Hx 168-Pin PGA Pinout — Signal Name Order (Sheet 2 of 2)

Signal Name	PGA Pin	Signal Name	PGA Pin	Signal Name	PGA Pin	Signal Name	PGA Pin
V_{CC}	J16	V_{CCPLL}	B10	V_{SS}	H3	V_{SS}	Q10
V_{CC}	K2	VOLDET	A5	V_{SS}	H15	V_{SS}	Q11
V_{CC}	K16	V_{SS}	A1	V_{SS}	J3	W/$\overline{R}$	S10
V_{CC}	M2	V_{SS}	C4	V_{SS}	J15	$\overline{WAIT}$	S12
V_{CC}	M16	V_{SS}	C7	V_{SS}	K3	$\overline{XINT0}$	B15
V_{CC}	N3	V_{SS}	C8	V_{SS}	K15	$\overline{XINT1}$	A15
V_{CC}	N15	V_{SS}	C9	V_{SS}	L3	$\overline{XINT2}$	A17
V_{CC}	Q6	V_{SS}	C10	V_{SS}	L15	$\overline{XINT3}$	B16
V_{CC}	R7	V_{SS}	C11	V_{SS}	M3	$\overline{XINT4}$	C15
V_{CC}	R8	V_{SS}	C12	V_{SS}	M15	$\overline{XINT5}$	B17
V_{CC}	R10	V_{SS}	F15	V_{SS}	Q7	$\overline{XINT6}$	C16
V_{CC}	R11	V_{SS}	G3	V_{SS}	Q8	$\overline{XINT7}$	C17
VCC5	C5	V_{SS}	G15	V_{SS}	Q9	--	--

Table 8. 80960Hx 168-Pin PGA Pinout — Pin Number Order (Sheet 1 of 2)

PGA Pin	Signal Name	PGA Pin	Signal Name	PGA Pin	Signal Name	PGA Pin	Signal Name
A1	V_{SS}	B14	NC	E15	V_{CC}	K15	V_{SS}
A2	$\overline{FAIL}$	B15	$\overline{XINT0}$	E16	A4	K16	V_{CC}
A3	DP0	B16	$\overline{XINT3}$	E17	A5	K17	A11
A4	DP2	B17	$\overline{XINT5}$	F1	D8	L1	D15
A5	VOLDET	C1	D3	F2	D6	L2	D14
A6	$\overline{TRST}$	C2	D1	F3	V_{CC}	L3	V_{SS}
A7	TDI	C3	$\overline{ONCE}$	F15	V_{SS}	L15	V_{SS}
A8	TDO	C4	V_{SS}	F16	V_{CC}	L16	A13
A9	NC	C5	VCC5	F17	A6	L17	A12
A10	NC	C6	V_{CC}	G1	D9	M1	D16
A11	CT0	C7	V_{SS}	G2	V_{CC}	M2	V_{CC}
A12	CT1	C8	V_{SS}	G3	V_{SS}	M3	V_{SS}
A13	CT2	C9	V_{SS}	G15	V_{SS}	M15	V_{SS}
A14	CT3	C10	V_{SS}	G16	A7	M16	V_{CC}
A15	$\overline{XINT1}$	C11	V_{SS}	G17	A8	M17	A14
A16	$\overline{RESET}$	C12	V_{SS}	H1	D11	N1	D17
A17	$\overline{XINT2}$	C13	CLKIN	H2	D10	N2	D18
B1	$\overline{BOFF}$	C14	V_{CC}	H3	V_{SS}	N3	V_{CC}
B2	STEST	C15	$\overline{XINT4}$	H15	V_{SS}	N15	V_{CC}
B3	DP1	C16	$\overline{XINT6}$	H16	V_{CC}	N16	A16
B4	DP3	C17	$\overline{XINT7}$	H17	A9	N17	A15
B5	TCK	D1	D5	J1	D12	P1	D19
B6	TMS	D2	D2	J2	V_{CC}	P2	D20
B7	V_{CC}	D3	NC	J3	V_{SS}	P3	D22
B8	$\overline{PCHK}$	D15	$\overline{NMI}$	J15	V_{SS}	P15	A20
B9	V_{CC}	D16	A2	J16	V_{CC}	P16	A19
B10	V_{CCPLL}	D17	A3	J17	A10	P17	A17
B11	V_{CC}	E1	D7	K1	D13	Q1	D21
B12	V_{CC}	E2	D4	K2	V_{CC}	Q2	D23
B13	NC	E3	D0	K3	V_{SS}	Q3	D26

Q6	V_{CC}	R1	D24	R13	BREQ	S8	$\overline{BLAST}$
Q7	V_{SS}	R2	D27	R14	A29	S9	$\overline{DEN}$
Q8	V_{SS}	R3	D31	R15	A26	S10	W/$\overline{R}$
Q9	V_{SS}	R4	$\overline{BTERM}$	R16	A23	S11	DT/$\overline{R}$
Q10	V_{SS}	R5	HOLD	R17	A22	S12	$\overline{WAIT}$
Q11	V_{SS}	R6	$\overline{ADS}$	S1	D25	S13	D/$\overline{C}$
Q12	$\overline{SUP}$	R7	V_{CC}	S2	D29	S14	$\overline{LOCK}$
Q13	A30	R8	V_{CC}	S3	$\overline{READY}$	S15	A31
Q14	A28	R9	$\overline{BE0}$	S4	HOLDA	S16	A27
Q15	A24	R10	V_{CC}	S5	$\overline{BE3}$	S17	A25

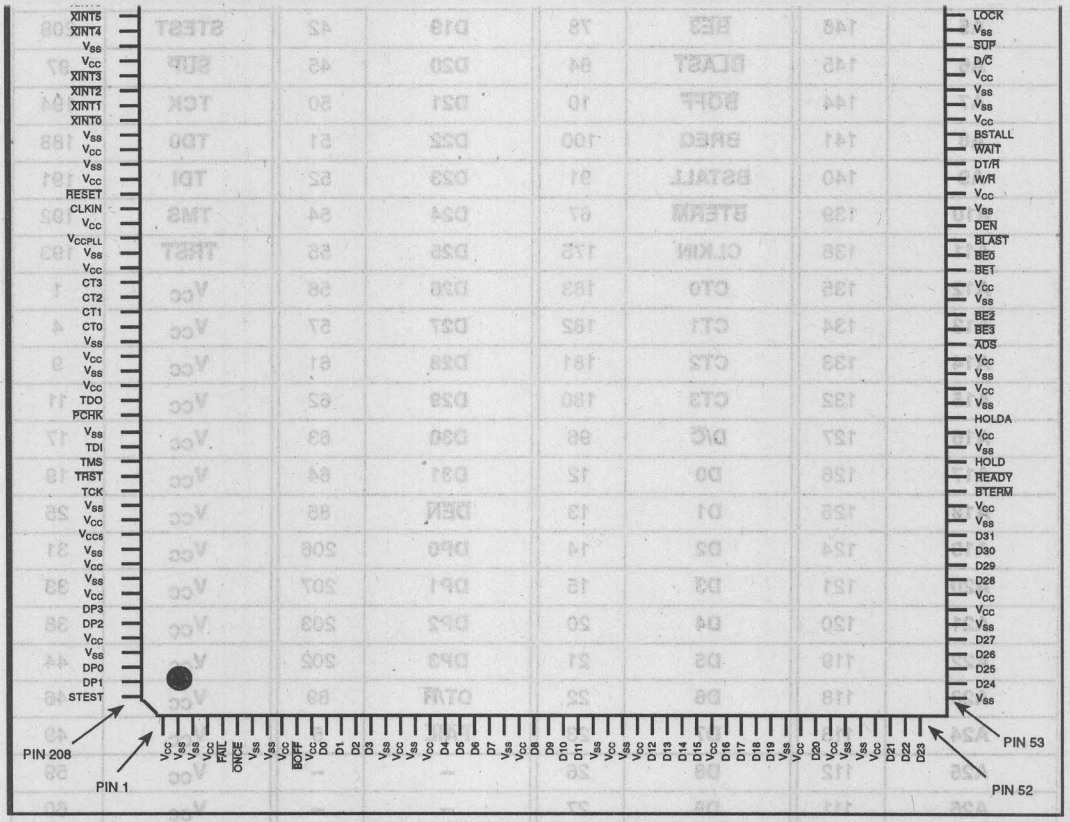


Figure 4. 80960Hx 208-Pin PQ2 Pinout

Table 9. 80960Hx PQ2 Pinout — Signal Name Order (Sheet 1 of 2)

Signal Name	PQ2 Pin	Signal Name	PQ2 Pin	Signal Name	PQ2 Pin	Signal Name	PQ2 Pin
A2	151	$\overline{BE0}$	83	D16	39	$\overline{PCHK}$	189
A3	150	$\overline{BE1}$	82	D17	40	$\overline{READY}$	68
A4	147	$\overline{BE2}$	79	D18	41	$\overline{RESET}$	174
A5	146	$\overline{BE3}$	78	D19	42	STEST	208
A6	145	$\overline{BLAST}$	84	D20	45	$\overline{SUP}$	97
A7	144	$\overline{BOFF}$	10	D21	50	TCK	194
A8	141	BREQ	100	D22	51	TD0	188
A9	140	BSTALL	91	D23	52	TDI	191
A10	139	$\overline{BTERM}$	67	D24	54	TMS	192
A11	138	CLKIN	175	D25	55	$\overline{TRST}$	193
A12	135	CT0	183	D26	56	V_{CC}	1
A13	134	CT1	182	D27	57	V_{CC}	4
A14	133	CT2	181	D28	61	V_{CC}	9
A15	132	CT3	180	D29	62	V_{CC}	11
A16	127	D/$\overline{C}$	96	D30	63	V_{CC}	17
A17	126	D0	12	D31	64	V_{CC}	19
A18	125	D1	13	$\overline{DEN}$	85	V_{CC}	25
A19	124	D2	14	DP0	206	V_{CC}	31
A20	121	D3	15	DP1	207	V_{CC}	33
A21	120	D4	20	DP2	203	V_{CC}	38
A22	119	D5	21	DP3	202	V_{CC}	44
A23	118	D6	22	DT/$\overline{R}$	89	V_{CC}	46
A24	113	D7	23	$\overline{FAIL}$	5	V_{CC}	49
A25	112	D8	26	--	--	V_{CC}	59
A26	111	D9	27	--	--	V_{CC}	60
A27	110	D10	28	--	--	V_{CC}	66
A28	107	D11	29	HOLD	69	V_{CC}	71
A29	106	D12	34	HOLDA	72	V_{CC}	74
A30	105	D13	35	$\overline{LOCK}$	99	V_{CC}	76
A31	104	D14	36	$\overline{NMI}$	159	V_{CC}	81
$\overline{ADS}$	77	D15	37	$\overline{ONCE}$	6	V_{CC}	87

Table 9. 80960Hx PQ2 Pinout — Signal Name Order (Sheet 2 of 2)

Signal Name	PQ2 Pin	Signal Name	PQ2 Pin	Signal Name	PQ2 Pin	Signal Name	PQ2 Pin
V_{CC}	92	V_{CC}	187	V_{SS}	70	V_{SS}	164
V_{CC}	95	V_{CC}	196	V_{SS}	73	V_{SS}	170
V_{CC}	101	V_{CC}	199	V_{SS}	75	V_{SS}	172
V_{CC}	102	V_{CC}	201	V_{SS}	80	V_{SS}	178
V_{CC}	109	V_{CC}	204	V_{SS}	86	V_{SS}	184
V_{CC}	115	VCC5	197	V_{SS}	93	V_{SS}	186
V_{CC}	117	V_{CCPLL}	177	V_{SS}	94	V_{SS}	190
V_{CC}	123	V_{SS}	2	V_{SS}	98	V_{SS}	195
V_{CC}	128	V_{SS}	3	V_{SS}	103	V_{SS}	198
V_{CC}	131	V_{SS}	7	V_{SS}	108	V_{SS}	200
V_{CC}	137	V_{SS}	8	V_{SS}	114	V_{SS}	205
V_{CC}	143	V_{SS}	16	V_{SS}	116	W/$\overline{R}$	88
V_{CC}	149	V_{SS}	18	V_{SS}	122	$\overline{WAIT}$	90
V_{CC}	153	V_{SS}	24	V_{SS}	129	$\overline{XINT0}$	169
V_{CC}	154	V_{SS}	30	V_{SS}	130	$\overline{XINT1}$	168
V_{CC}	158	V_{SS}	32	V_{SS}	136	$\overline{XINT2}$	167
V_{CC}	165	V_{SS}	43	V_{SS}	142	$\overline{XINT3}$	166
V_{CC}	171	V_{SS}	47	V_{SS}	148	$\overline{XINT4}$	163
V_{CC}	173	V_{SS}	48	V_{SS}	152	$\overline{XINT5}$	162
V_{CC}	176	V_{SS}	53	V_{SS}	155	$\overline{XINT6}$	161
V_{CC}	179	V_{SS}	58	V_{SS}	156	$\overline{XINT7}$	160
V_{CC}	185	V_{SS}	65	V_{SS}	157	--	--

Table 10. 80960Hx PQ2 Pinout — Pin Number Order (Sheet 1 of 2)

PQ2 Pin	Signal Name	PQ2 Pin	Signal Name	PQ2 Pin	Signal Name	PQ2 Pin	Signal Name
1	V_{CC}	31	V_{CC}	61	D28	91	BSTALL
2	V_{SS}	32	V_{SS}	62	D29	92	V_{CC}
3	V_{SS}	33	V_{CC}	63	D30	93	V_{SS}
4	V_{CC}	34	D12	64	D31	94	V_{SS}
5	$\overline{FAIL}$	35	D13	65	V_{SS}	95	V_{CC}
6	$\overline{ONCE}$	36	D14	66	V_{CC}	96	D/$\overline{C}$
7	V_{SS}	37	D15	67	$\overline{BTERM}$	97	$\overline{SUP}$
8	V_{SS}	38	V_{CC}	68	$\overline{READY}$	98	V_{SS}
9	V_{CC}	39	D16	69	HOLD	99	$\overline{LOCK}$
10	$\overline{BOFF}$	40	D17	70	V_{SS}	100	BREQ
11	V_{CC}	41	D18	71	V_{CC}	101	V_{CC}
12	D0	42	D19	72	HOLDA	102	V_{CC}
13	D1	43	V_{SS}	73	V_{SS}	103	V_{SS}
14	D2	44	V_{CC}	74	V_{CC}	104	A31
15	D3	45	D20	75	V_{SS}	105	A30
16	V_{SS}	46	V_{CC}	76	V_{CC}	106	A29
17	V_{CC}	47	V_{SS}	77	$\overline{ADS}$	107	A28
18	V_{SS}	48	V_{SS}	78	$\overline{BE3}$	108	V_{SS}
19	V_{CC}	49	V_{CC}	79	$\overline{BE2}$	109	V_{CC}
20	D4	50	D21	80	V_{SS}	110	A27
21	D5	51	D22	81	V_{CC}	111	A26
22	D6	52	D23	82	$\overline{BE1}$	112	A25
23	D7	53	V_{SS}	83	$\overline{BE0}$	113	A24
24	V_{SS}	54	D24	84	$\overline{BLAST}$	114	V_{SS}
25	V_{CC}	55	D25	85	$\overline{DEN}$	115	V_{CC}
26	D8	56	D26	86	V_{SS}	116	V_{SS}
27	D9	57	D27	87	V_{CC}	117	V_{CC}
28	D10	58	V_{SS}	88	W/$\overline{R}$	118	A23
29	D11	59	V_{CC}	89	DT/$\overline{R}$	119	A22
30	V_{SS}	60	V_{CC}	90	$\overline{WAIT}$	120	A21

Table 10. 80960Hx PQ2 Pinout — Pin Number Order (Sheet 2 of 2)

PQ2 Pin	Signal Name	PQ2 Pin	Signal Name	PQ2 Pin	Signal Name	PQ2 Pin	Signal Name
121	A20	143	V_{CC}	165	V_{CC}	187	V_{CC}
122	V_{SS}	144	A7	166	$\overline{XINT3}$	188	TD0
123	V_{CC}	145	A6	167	$\overline{XINT2}$	189	$\overline{PCHK}$
124	A19	146	A5	168	$\overline{XINT1}$	190	V_{SS}
125	A18	147	A4	169	$\overline{XINT0}$	191	TDI
126	A17	148	V_{SS}	170	V_{SS}	192	TMS
127	A16	149	V_{CC}	171	V_{CC}	193	$\overline{TRST}$
128	V_{CC}	150	A3	172	V_{SS}	194	TCK
129	V_{SS}	151	A2	173	V_{CC}	195	V_{SS}
130	V_{SS}	152	V_{SS}	174	$\overline{RESET}$	196	V_{CC}
131	V_{CC}	153	V_{CC}	175	CLKIN	197	VCC5
132	A15	154	V_{CC}	176	V_{CC}	198	V_{SS}
133	A14	155	V_{SS}	177	V_{CCPLL}	199	V_{CC}
134	A13	156	V_{SS}	178	V_{SS}	200	V_{SS}
135	A12	157	V_{SS}	179	V_{CC}	201	V_{CC}
136	V_{SS}	158	V_{CC}	180	CT3	202	DP3
137	V_{CC}	159	$\overline{NMI}$	181	CT2	203	DP2
138	A11	160	$\overline{XINT7}$	182	CT1	204	V_{CC}
139	A10	161	$\overline{XINT6}$	183	CT0	205	V_{SS}
140	A9	162	$\overline{XINT5}$	184	V_{SS}	206	DP0
141	A8	163	$\overline{XINT4}$	185	V_{CC}	207	DP1
142	V_{SS}	164	V_{SS}	186	V_{SS}	208	STEST

3.3 Package Thermal Specifications

The 80960Hx is specified for operation when T_C (case temperature) is within the range of 0°C–85°C. T_C may be measured in any environment to determine whether the 80960Hx is within the specified operating range. Measure the case temperature at the center of the top surface, opposite the pins. Refer to Figure 5.

T_A (ambient temperature) is calculated from θ_{CA} (thermal resistance from case to ambient) using the equation:

$$T_A = T_C - P^*\theta_{CA}$$

Table 11 shows the maximum T_A allowable (without exceeding T_C) at various airflows and operating frequencies (f_{CLKIN}).

Note that T_A is greatly improved by attaching fins or a heatsink to the package. P (maximum power consumption) is calculated by using the typical I_{CC} as tabulated in Section 4.5, DC Specifications (pg. 29) and V_{CC} of 3.3 V.

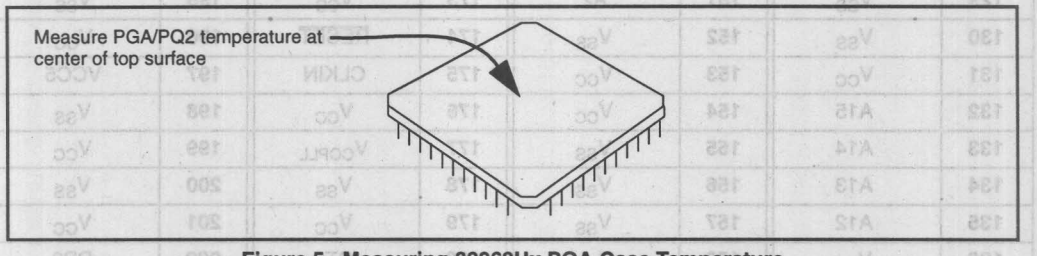


Figure 5. Measuring 80960Hx PGA Case Temperature

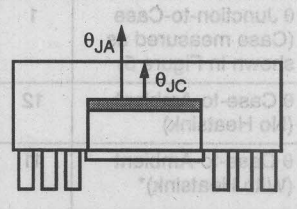
Table 11. Maximum T_A at Various Airflows in °C (PGA Package Only)

			Airflow-ft/min (m/sec)					
		f_{CLKIN} (MHz)	0 (0)	200 (1.01)	400 (2.03)	600 (3.04)	800 (4.06)	1000 (5.07)
Core 1X Bus Clock	T_A with Heatsink*	40	52	62	70	72	75	75
		33	56	65	72	74	77	77
		25	62	69	75	76	78	78
	T_A without Heatsink	40	41	49	57	62	65	67
		33	47	54	61	65	68	70
		25	55	60	66	69	71	73
Core 2X Bus Clock	T_A with Heatsink*	33	36	51	63	67	70	70
		25	46	58	67	70	73	73
		16	58	67	73	75	77	77
	T_A without Heatsink	33	21	32	44	51	55	59
		25	34	43	52	58	61	64
		16	50	56	63	67	69	71
Core 3X Bus Clock	T_A with Heatsink*	25	32	49	61	65	69	69
		20	42	55	65	69	72	72
	T_A without Heatsink	25	16	28	40	49	53	57
		20	28	38	48	55	59	62

NOTES: *0.285" high unidirectional heatsink (Al alloy 6061, 50 mil fin width, 150 mil center-to-center fin spacing).

Table 12. 80960Hx 168-Pin PGA Package Thermal Characteristics

Parameter	Thermal Resistance — °C/Watt: Airflow — ft./min (m/sec)					
	0 (0)	200 (1.01)	400 (2.03)	600 (3.07)	800 (4.06)	1000 (5.07)
θ Junction-to-Case (Case measured as shown in Figure 5)	1.5	1.5	1.5	1.5	1.5	1.5
θ Case-to-Ambient (No Heatsink)	17	14	11	9	8	7
θ Case-to-Ambient (With Heatsink)*	13	9	6	5	4	4



NOTES:

1. This table applies to 80960Hx PGA plugged into socket or soldered directly to board.
2. $\theta_{JA} = \theta_{JC} + \theta_{CA}$

*0.285" high unidirectional heatsink (Al alloy 6061, 50 mil fin width, 150 mil center-to-center fin spacing).

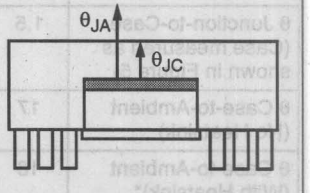
Table 13. Maximum T_A at Various Airflows in °C (PQ2 Package Only)

			Airflow-ft/min (m/sec)					
		f_{CLKIN} (MHz)	0 (0)	200 (1.01)	400 (2.03)	600 (3.04)	800 (4.06)	1000 (5.07)
Core 1X Bus Clock	T_A with Heatsink*	40	57	67	72	72	75	75
		33	61	70	74	74	77	77
		25	66	73	76	76	78	78
	T_A without Heatsink	40	54	59	65	65	67	67
		33	59	63	68	68	70	70
		25	64	67	71	71	73	73
Core 2X Bus Clock	T_A with Heatsink*	33	44	59	67	67	70	70
		25	52	64	70	70	73	73
		16	63	71	75	75	77	77
	T_A without Heatsink	33	40	48	55	55	59	59
		25	49	55	61	61	64	64
		16	61	65	69	69	71	71
Core 3X Bus Clock	T_A with Heatsink*	25	40	57	65	65	69	69
		20	48	62	69	69	72	72
	T_A without Heatsink	25	36	45	53	53	57	57
		20	45	52	59	59	62	62

NOTES: *0.285" high unidirectional heatsink (Al alloy 6061, 50 mil fin width, 150 mil center-to-center fin spacing).

Table 14. 80960Hx 208-Pin PQ2 Package Thermal Characteristics

Parameter	Thermal Resistance — °C/Watt: Airflow — ft./min (m/sec)					
	0 (0)	200 (1.01)	400 (2.03)	600 (3.07)	800 (4.06)	1000 (5.07)
θ Junction-to-Case (Case measured as shown in Figure 5)	1	1	1	1	1	1
θ Case-to-Ambient (No Heatsink)	12	10	8	8	7	7
θ Case-to-Ambient (With Heatsink)*	11	7	5	5	4	4



NOTES:

1. This table applies to 80960Hx PQ2 plugged into socket or soldered directly to board.
2. $\theta_{JA} = \theta_{JC} + \theta_{CA}$

*0.285" high unidirectional heatsink (Al alloy 6061, 50 mil fin width, 150 mil center-to-center fin spacing).

3.4 Heat Sink Adhesives

Intel recommends silicone-based adhesives to attach heat sinks to the PGA package. There is no particular recommendation concerning the PQ2 package.

3.5 PowerQuad2 Plastic Package

The 80960Hx family is available in an improved version of the common 208-lead SQFP plastic package called the PowerQuad2* (PQ2). The PQ2 package dimensions and lead pitch are identical to the SQFP package, so the PQ2 fits into the same board footprint. The advantage of the PQ2 package is the superior thermal conductivity that allows the plastic version of the 80960Hx to operate with the same 0 - 85°C temperature specifications as the more expensive ceramic PGA package.

The PQ2 package integrates a copper heat sink within the package to dissipate heat effectively. See Table 13 and Table 14 on page 24.

3.6 Stepping Register Information

The memory-mapped register at FF008710H contains the 80960Hx Device ID. The ID is identical to the ID obtained from a JTAG Query. Figure 6 defines the current 80960Hx Device IDs. The value for device identification is compliant with the IEEE 1149.1 specification and Intel standards. Table 15 describes the fields of the device ID.

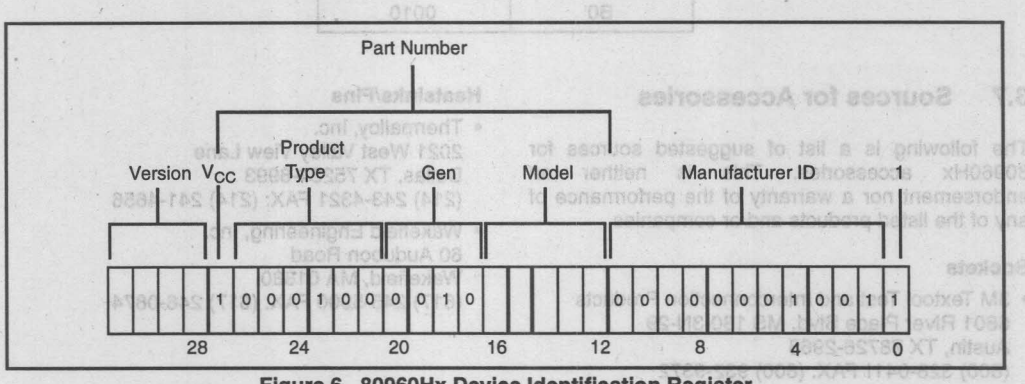


Figure 6. 80960Hx Device Identification Register

Table 15. Fields of 80960Hx Device ID

Field	Value	Definition
Version	See Table 17	Indicates major stepping changes.
V_{CC}	1 = 3.3 V device	Indicates that a device is 3.3 V.
Product Type	00 0100 (Indicates i960 CPU)	Designates type of product.
Generation Type	0010 = H-series	Indicates the generation (or series) the product belongs to.
Model	See Table 16	Indicates member within a series and specific model information.
Manufacturer ID	000 0000 1001 (Indicates Intel)	Manufacturer ID assigned by IEEE.

Table 16. 80960Hx Device ID Model Types

Device	Version	V_{CC}	Product	Gen.	Model	Manufacturer ID	'1'
80960HA	See Table 17	1	000100	0010	00000	00000001001	1
80960HD		1	000100	0010	00001	00000001001	1
80960HT		1	000100	0010	00010	00000001001	1

Table 17. Device ID Version Numbers for Different Steppings

Stepping	Version
A0	0000
A1	0001
A2	0001
B0	0010

3.7 Sources for Accessories

The following is a list of suggested sources for 80960Hx accessories. This is neither an endorsement nor a warranty of the performance of any of the listed products and/or companies.

Sockets

- 3M Textool Test and Interconnection Products
  6801 River Place Blvd. MS 130-3N-29
  Austin, TX 78726-2963
  (800) 328-0411 FAX: (800) 932-9372
- Augat, Inc. Interconnection Products Group
  452 John Dietsch Blvd. P.O. Box 2510
  Attleboro Falls, MA 02763
  (508) 699-7646
- Concept Mfg, Inc. (Decoupling Sockets)
  400 Walnut St. Suite 609
  Redwood City, CA 94063
  (415) 365-1162 FAX: (415) 365-1164

Heatsinks/Fins

- Thermalloy, Inc.
  2021 West Valley View Lane
  Dallas, TX 75234-8993
  (214) 243-4321 FAX: (214) 241-4656
- Wakefield Engineering, Inc.
  60 Audubon Road
  Wakefield, MA 01880
  (617) 245-5900 FAX: (617) 246-0874

4.0 ELECTRICAL SPECIFICATIONS

4.1 Absolute Maximum Ratings

Parameter	Maximum Rating
Storage Temperature	–65°C to +150°C
Case Temperature Under Bias	–65°C to +110°C
Supply Voltage wrt. V_{SS}	–0.5 V to + 4.6 V
Voltage on VCC5 wrt. V_{SS}	–0.5 V to + 6.5 V
Voltage on Other Pins wrt. V_{SS}	–0.5 V to V_{CC} + 0.5 V

NOTICE: This document contains information on products in the sampling and initial production phases of development. The specifications are subject to change without notice. Verify with your local Intel sales office that you have the latest datasheet before finalizing a design.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

4.2 Operating Conditions

Table 18. Operating Conditions

Symbol	Parameter	Min	Max	Units
V_{CC}	Supply Voltage	3.15	3.45	V
VCC5	Input Protection Bias	3.15	5.5	V
f_{CLKIN} 1xcore	Input Clock Frequency - 1x Core (80960HA)	16	40	MHz
f_{CLKIN} 2xcore	Input Clock Frequency - 2x Core (80960HD)	16	33	MHz
f_{CLKIN} 3xcore	Input Clock Frequency - 3x Core (80960HT)	16	25	MHz
T_C	Case Temp Under Bias PGA Pkg	0	85	°C

4.3 Recommended Connections

Power and ground connections must be made to multiple V_{CC} and V_{SS} (GND) pins. Every 80960Hx-based circuit board should include power (V_{CC}) and ground (V_{SS}) planes for power distribution. Every V_{CC} pin must be connected to the power plane; every V_{SS} pin must be connected to the ground plane. Pins identified as "NC" —no connect pins— **must not** be connected in the system.

Liberal decoupling capacitance should be placed near the 80960Hx. The processor can cause transient power surges when its numerous output buffers transition, particularly when connected to large capacitive loads.

Low inductance capacitors and interconnects are recommended for best high-frequency electrical performance. Inductance can be reduced by shortening the board traces between the processor and decoupling capacitors as much as possible. Capacitors specifically designed for PGA packages offer the lowest possible inductance.

For reliable operation, always connect unused inputs to an appropriate signal level. In particular, any unused interrupt ($\overline{XINT7:0}$, $\overline{NMI}$) input should be connected to V_{CC} through a pull-up resistor, as should $\overline{BTERM}$ if not used. Pull-up resistors should be in the in the range of 20 KΩ for each pin tied high. If $\overline{READY}$ or HOLD are not used, the unused input should be connected to ground. **N.C. pins must always remain unconnected.**

4.4 VCC5 Pin Requirements (V_{DIFF})

In mixed voltage systems that drive 80960Hx processor inputs in excess of 3.3 V, the VCC5 pin must be connected to the system's 5 V supply. To limit current flow into the VCC5 pin, there is a limit to the voltage differential between the VCC5 pin and the other V_{CC} pins. The voltage differential between the 80960Hx VCC5 pin and its 3.3 V V_{CC} pins should never exceed 2.25 V. This limit applies to power-up, power-down, and steady-state operation. Table 19 outlines this requirement.

Meeting this requirement ensures proper operation and guarantees that the current draw into the VCC5 pin does not exceed the ICC5 specification.

If the voltage difference requirements cannot be met due to system design limitations, an alternate solution may be employed. As shown in Figure 7, a minimum of 100 Ω series resistor may be used to limit the current into the VCC5 pin. This resistor ensures that current drawn by the VCC5 pin does not exceed the maximum rating for this pin.

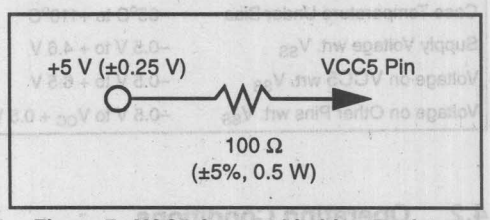


Figure 7. VCC5 Current-Limiting Resistor

This resistor is not necessary in systems that can guarantee the V_{DIFF} specification.

In 3.3 V-only systems and systems that drive 80960Hx pins from 3.3 V logic, connect the VCC5 pin directly to the 3.3 V V_{CC} plane.

Table 19. V_{DIFF} Specification for Dual Power Supply Requirements (3.3 V, 5 V)

Symbol	Parameter	Min	Max	Units	Notes
V_{DIFF}	VCC5-V_{CC} Difference		2.25	V	VCC5 input should not exceed V_{CC} by more than 2.25 V during power-up and power-down, or during steady-state operation.

4.5 DC Specifications

Table 20. 80960Hx DC Characteristics (Sheet 1 of 2)

Per the conditions described in Section 4.3, Recommended Connections (pg. 27).

Symbol	Parameter		Min	Typ	Max	Units	Notes
V_{IL}	Input Low Voltage		– 0.3		+0.8	V	
V_{IH}	Input High Voltage		2.0		VCC5 + 0.3	V	
V_{OL}	Output Low Voltage All outputs except $\overline{FAIL}$				0.4 0.2	V	I_{OL} = 3 mA I_{OL} = 100 µA
V_{OL}	Output Low Voltage $\overline{FAIL}$ pin				0.4	V	I_{OL} = 5 mA
V_{OH}	Output High Voltage		2.4 V_{CC} – 0.2			V V	I_{OH} = –3 mA I_{OH} = – 100 µA
I_{LI}	Input Leakage Current Non-Test Inputs TDI, TMS, $\overline{TRST}$ and $\overline{ONCE}$ pins		-1		1 -110	µA µA	$0 \le V_{IN} \le V_{CC}$ V_{IN} = 0 V
I_{LO}	Output Leakage Current Non-Test Outputs TDO pin				1 5	µA µA	$0.45 \le V_{OUT} \le V_{CC}$ $0.45 \le V_{OUT} \le V_{CC}$
I_{CC} Active (Power Supply)	80960HA	25			790	mA	(4,5)
		33			1000		
		40			1180		
	80960HD	32			950		
		50			1390		
		66			1790		
	80960HT	60			1350		
		75			1650		
I_{CC} Active (Thermal)	80960HA	25		550		mA	(4,6)
		33		680			
		40		790			
	80960HD	32		630			
		50		910			
		66		1150			
	80960HT	60		1020			
		75		1240			

Table 20. 80960Hx DC Characteristics (Sheet 2 of 2)

Per the conditions described in Section 4.3, Recommended Connections (pg. 27).

Symbol	Parameter	Min	Typ	Max	Units	Notes
I_{CC} Test (Reset Mode)	80960HA 25			540	mA	(7,8)
	33			710		
	40			850		
	80960HD 32			640		
	50			990		
	66			1300		
	80960HT 60			1150		
	75			1430		
I_{CC} Test (ONCE mode)				25	mA	(7)
ICC5 Current on the VCC5 Pin	80960HA			200	µA	(9)
	80960HD			200		
	80960HT			200		
C_{IN}	Input Capacitance for: PQ2			12	pF	
	PGA			12	pF	F_C = 1 MHz
C_{OUT}	Output Capacitance of each output pin			12	pF	F_C = 1 MHz (3)
$C_{I/O}$	I/O Pin Capacitance			12	pF	F_C = 1 MHz
R_{PU}	Internal Pull-Up Resistance for ONCE, TMS, TDI and TRST	30	65	100	kΩ	

NOTES:

1. I_{CC} Maximum is measured at worst case frequency, V_{CC}, and temperature, with device operating and outputs loaded to the test conditions described in Section 4.6.1, AC Test Conditions (pg. 34).
2. I_{CC} Typical is not tested.
3. Output Capacitance is the capacitive load of a floating output.
4. Measured with device operating and outputs loaded to the test conditions in Figure 8, AC Test Load (pg. 32). Input signals rise to V_{CC} and fall to V_{SS}.
5. I_{CC} Active (Power Supply) value is provided for selecting your system's power supply. It is measured using one of the worst case instruction mixes with V_{CC} = 3.45 V. This parameter is characterized but not tested.
6. I_{CC} Active (Thermal) value is provided for your system's thermal management. Typical I_{CC} is measured with V_{CC} = 3.3 V and temperature = 25°C. This parameter is characterized but not tested.
7. I_{CC} Test (Power modes) refers to the I_{CC} values that are tested when the 80960HA/HD/HT is in Reset mode or ONCE mode with V_{CC} = 3.45 V.
8. Worst case is V_{CC} = 3.45 V, 0°C.
9. ICC5 is tested at V_{CC} = 3.0 V, VCC5 = 5.25 V.

4.6 AC Specifications

Table 21. 80960Hx AC Characteristics (Sheet 1 of 2)

Per the conditions in Section 4.2, Operating Conditions (pg. 27) and Section 4.6.1, AC Test Conditions (pg. 34).

Symbol	Parameter		Min	Max	Units	Notes
Input Clock (1,7)						
T_F	CLKIN Frequency	80960HA	16.0	40.0	MHz	
		80960HD	16.0	33.33	MHz	
		80960HT	16.0	25.0	MHz	
T	CLKIN Period	80960HA	25.0	62.5	ns	
		80960HD	30.0	62.5	ns	
		80960HT	40.0	62.5	ns	
T_{CS}	CLKIN Period Stability		-250.0	+250.0	ps	(11)
T_{CH}	CLKIN High Time		8.0		ns	(11)
T_{CL}	CLKIN Low Time	80960HA	8.0		ns	(11)
		80960HD	8.0		ns	
		80960HT	8.0		ns	
T_{CR}	CLKIN Rise Time		0.0	4.0	ns	(11)
T_{CF}	CLKIN Fall Time		0.0	4.0	ns	(11)
Synchronous Outputs (1,2,3,6)						
T_{OV1}, T_{OH1}	Output Valid Delay and Output Hold for all outputs except DT/$\overline{R}$, $\overline{BLAST}$, and BREQ for 3.3 V inputs and IO's.		1.5	9.5	ns	(13)
	Same as above, but for 5.5 V inputs and I/O's.		1.5	12.5	ns	
T_{OV2}, T_{OH2}	Output Valid Delay and Output Hold for DT/$\overline{R}$		T/2 + 1.5	T/2 + 9.5	ns	
T_{OV3}, T_{OH3}	Output Valid Delay and Output Hold for $\overline{BLAST}$		1.5	9.0	ns	
T_{OV4}, T_{OH4}	Output Valid Delay and Output Hold for BREQ		0.5	9.0	ns	
T_{OV5}, T_{OH5}	Output Valid Delay and Output Hold for A3:2		1.5	8.5		
T_{OF}	Output Float for all outputs		1.5	9.0	ns	(11)
Synchronous Inputs (1,7,8,9)						
T_{IS1}	Input Setup for all inputs except $\overline{READY}$, $\overline{BTERM}$, HOLD, and $\overline{BOFF}$		2.5		ns	
T_{IH1}	Input Hold for all inputs except $\overline{READY}$, $\overline{BTERM}$, HOLD, and $\overline{BOFF}$		2.5		ns	
T_{IS2}	Input Setup for $\overline{READY}$, $\overline{BTERM}$, HOLD, and $\overline{BOFF}$		6.0		ns	
T_{IH2}	Input Hold for $\overline{READY}$, $\overline{BTERM}$, HOLD, and $\overline{BOFF}$		2.5		ns	
Relative Output Timings (1,2,3,6,10)						
T_{AVSH1}	A31:2 Valid to $\overline{ADS}$ Rising		T – 5	T + 5	ns	(10)
T_{AVSH2}	$\overline{BE3:0}$, W/$\overline{R}$, $\overline{SUP}$, D/$\overline{C}$ Valid to $\overline{ADS}$ Rising		T – 5	T + 5	ns	(10)

NOTES:
See Table 22, AC Characteristics Notes (pg. 33) for all notes related to AC specifications.

T_{NLQV}	WAIT Falling to Output Data Valid	-5.0	5.0	ns	(10)
T_{DVNH}	Output Data Valid to WAIT Rising	-5.0 + N*T	5.0 + N*T	ns	(4), (10)
T_{NLNH}	WAIT Falling to WAIT Rising	-4.0 + N*T	4.0 + N*T	ns	(4), (10)
T_{NHQX}	Output Data Hold after WAIT Rising	-5.0 + (N+1)*T	5.0 + (N+1)*T	ns	(5), (10)
T_{EHTV}	DT/R Hold after DEN High	T/2 – 5	Infinite	ns	(10)
T_{TVEL}	DT/R Valid to DEN Falling	T/2 – 4		ns	(10)
	Relative Input Timings (1,7,10)				
T_{IS7}	XINT7:0, NMI Input Setup	6.0		ns	(9)
T_{IH7}	XINT7:0, NMI Input Hold	2.5		ns	(9)
T_{IS8}	RESET Input Setup	3.0		ns	(8)
T_{IH8}	RESET Input Hold	T/4 + 1		ns	(8)

NOTES:
See Table 22, AC Characteristics Notes (pg. 33) for all notes related to AC specifications.

Table 22. AC Characteristics Notes

NOTES:

1. See Section 4.6.2, AC Timing Waveforms (pg. 34) for waveforms and definitions.
2. See Figure 25, Output Delay or Hold vs. Load Capacitance (pg. 42) for capacitive derating information for output delays and hold times.
3. See Figure 22, Rise and Fall Time Derating at 85°C and Minimum VCC (pg. 41) for capacitive derating information for rise and fall times.
4. Where N is the number of N_{RAD}, N_{RDD}, N_{WAD} or N_{WDD} wait states that are programmed in the Bus Controller Region Table. $\overline{WAIT}$ never goes active when there are no wait states in an access.
5. N = Number of wait states inserted with $\overline{READY}$.
6. These specifications are guaranteed by the processor.
7. These specifications must be met by the system for proper operation of the processor.
8. $\overline{RESET}$ is an asynchronous input that has no required setup and hold time for proper operation. However, to guarantee the device exits the reset mode synchronized to a particular clock edge, the rising edge of $\overline{RESET}$ must meet setup and hold times to the rising edge of the CLKIN.
9. The interrupt pins are synchronized internally by the 80960Hx. They have no required setup or hold times for proper operation. These pins are sampled by the interrupt controller every clock and must be active for at least two consecutive CLKIN rising edges when asserting them asynchronously. To guarantee recognition at a particular clock edge, the setup and hold times shown must be met.
10. Relative Output timings are not tested.
11. Not tested.
12. The processor minimizes changes to the bus signals when transitioning from a bus cycle to an idle bus for the following signals: A31:4, SUP, CT3:0, D/$\overline{C}$, $\overline{LOCK}$, W/$\overline{R}$, BE3:0.
13. Worst-case T_{OV} condition occurs on I/O pins when pins transition from a floating high input to driving a low output state. The Data Bus pins encounter this condition during a read followed by a write. 5 V signals take 3 ns longer to discharge than 3.3 V signals at 50 pF loads.

Table 23. 80960Hx Boundary Scan Test Signal Timings (Sheet 1 of 2)

Symbol	Parameter	Min	Max	Units	Notes
T_{BSF}	TCK Frequency	0	8	MHz	
T_{BSC}	TCK Period	125	Infinite	ns	
T_{BSCH}	TCK High Time	40		ns	Measured at 1.5 V (1)
T_{BSCL}	TCK Low Time	40		ns	Measured at 1.5 V (1)
T_{BSCR}	TCK Rise Time		8	ns	0.8 V to 2.0 V (1)
T_{BSCF}	TCK Fall Time		8	ns	2.0 V to 0.8 V (1)
T_{BSIS1}	Input Setup to TCK — TDI, TMS	8		ns	
T_{BSIH1}	Input Hold from TCK — TDI, TMS	10		ns	
T_{BSOV1}	TDO Valid Delay	3	30	ns	

NOTES:

1. Not tested.

Table 23. 80960Hx Boundary Scan Test Signal Timings (Sheet 2 of 2)

Symbol	Parameter	Min	Max	Units	Notes
T_{BSOF1}	TDO Float Delay		36	ns	(1)
T_{BSOV2}	All Outputs (Non-Test) Valid Delay	3	30	ns	Relative to TCK
T_{BSOF2}	All Outputs (Non-Test) Float Delay		36	ns	Relative to TCK (1)
T_{BSIS2}	Input Setup to TCK - All Inputs (Non-Test)	8		ns	
T_{BSIH2}	Input Hold from TCK - All Inputs (Non-Test)	10		ns	

NOTES:

1. Not tested.

4.6.1 AC Test Conditions

AC values are derived using the 50 pF load shown in Figure 8. Figure 25, Output Delay or Hold vs. Load Capacitance (pg. 42), shows how timings vary with load capacitance. Input waveforms (except for CLKIN) are assumed to have a rise and fall time of ≤ 2 ns from 0.8 V to 2.0 V.

4.6.2 AC Timing Waveforms

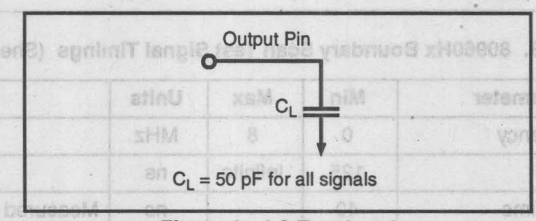


Figure 8. AC Test Load

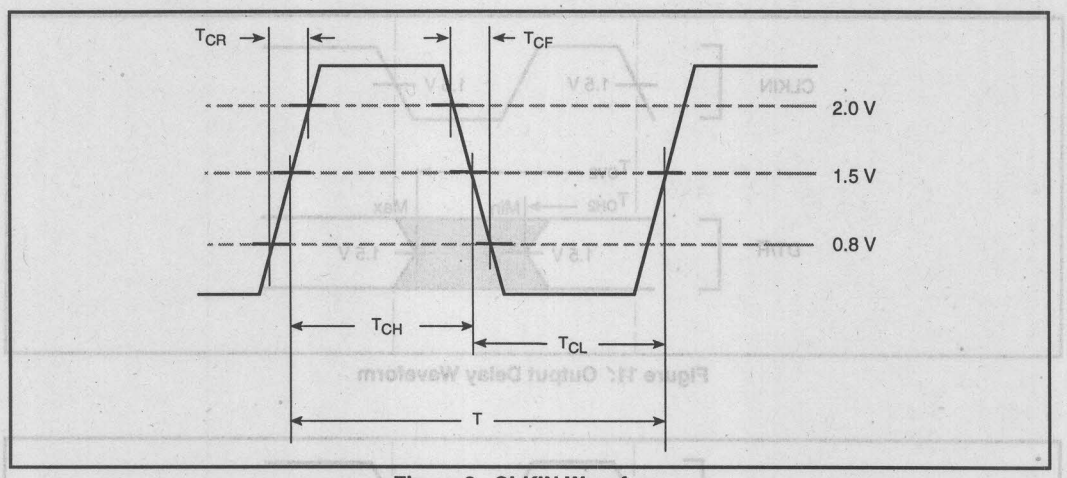


Figure 9. CLKIN Waveform

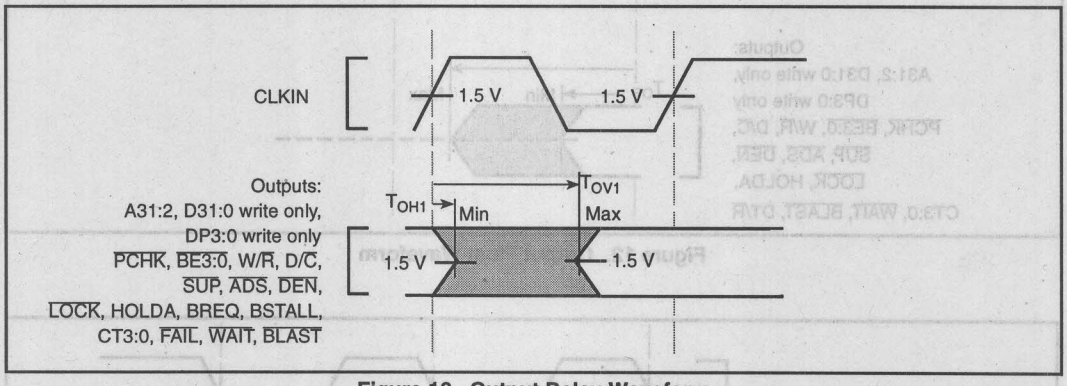


Figure 10. Output Delay Waveform

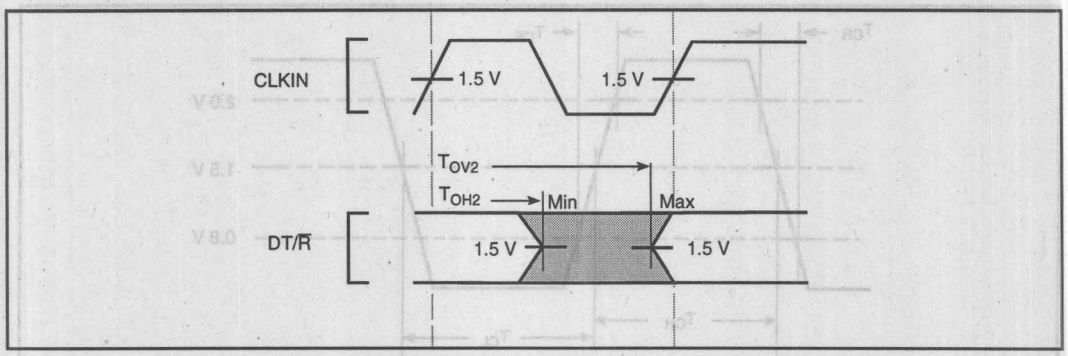


Figure 11. Output Delay Waveform

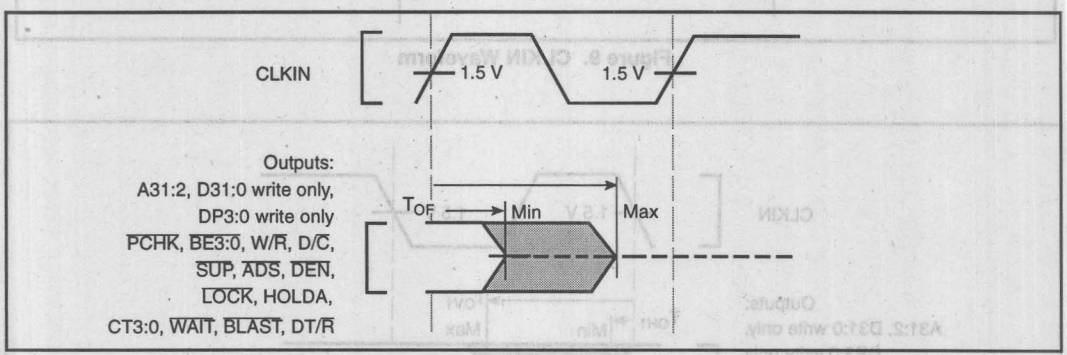


Figure 12. Output Float Waveform

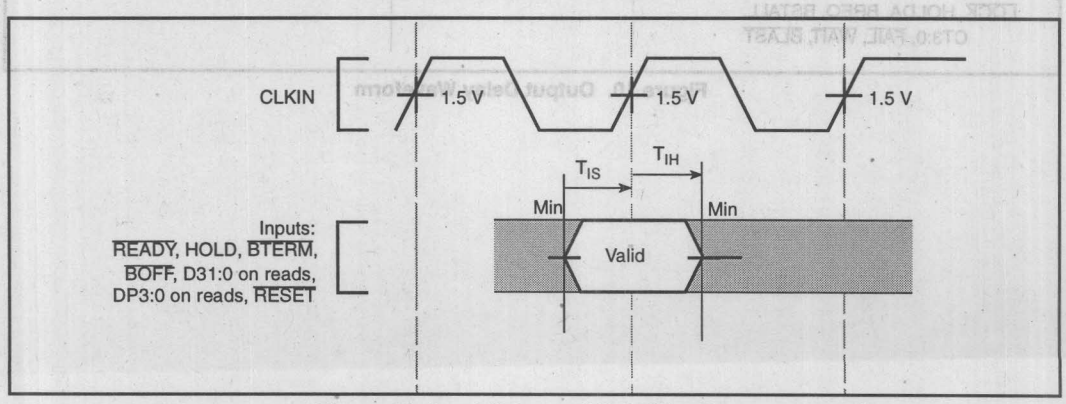
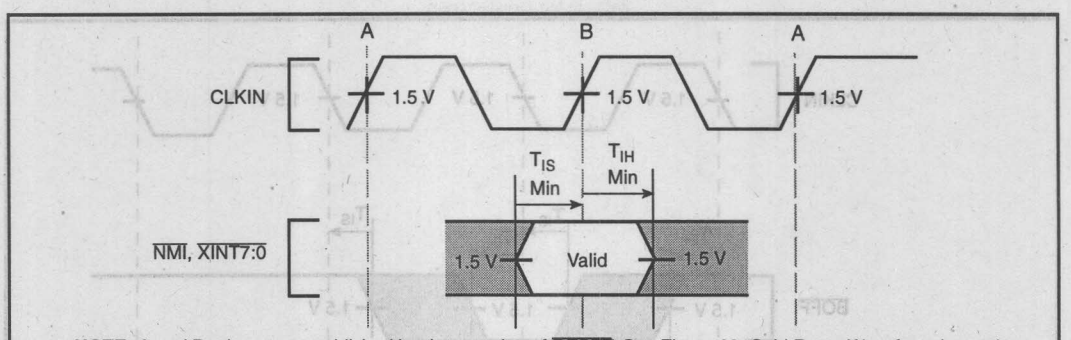
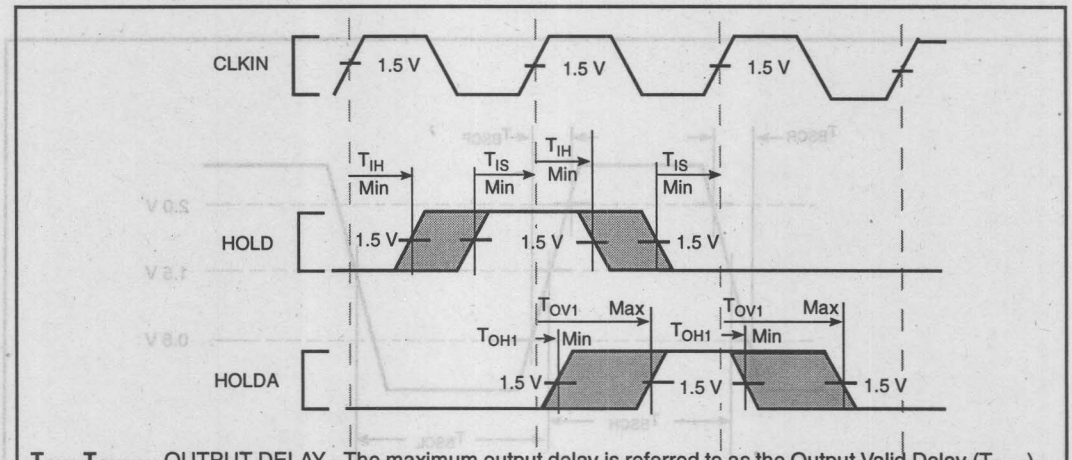


Figure 13. Input Setup and Hold Waveform



NOTE: A and B edges are established by de-assertion of RESET. See Figure 29, Cold Reset Waveform (pg. 45).

Figure 14. NMI, XINT7:0 Input Setup and Hold Waveform



T_{OV1} T_{OH1} — OUTPUT DELAY - The maximum output delay is referred to as the Output Valid Delay (T_{OV1}). The minimum output delay is referred to as the Output Hold (T_{OH1}).

T_{IS} T_{IH} — INPUT SETUP AND HOLD - The input setup and hold requirements specify the sampling window during which synchronous inputs must be stable for correct processor operation.

Figure 15. Hold Acknowledge Timings

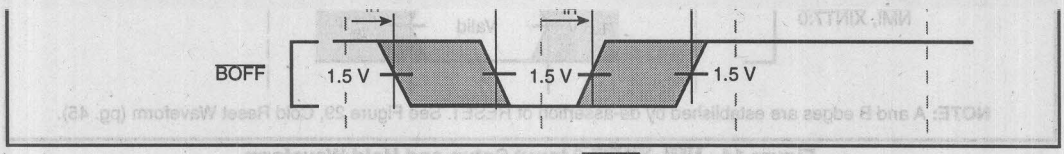


Figure 16. Bus Backoff ($\overline{BOFF}$) Timings

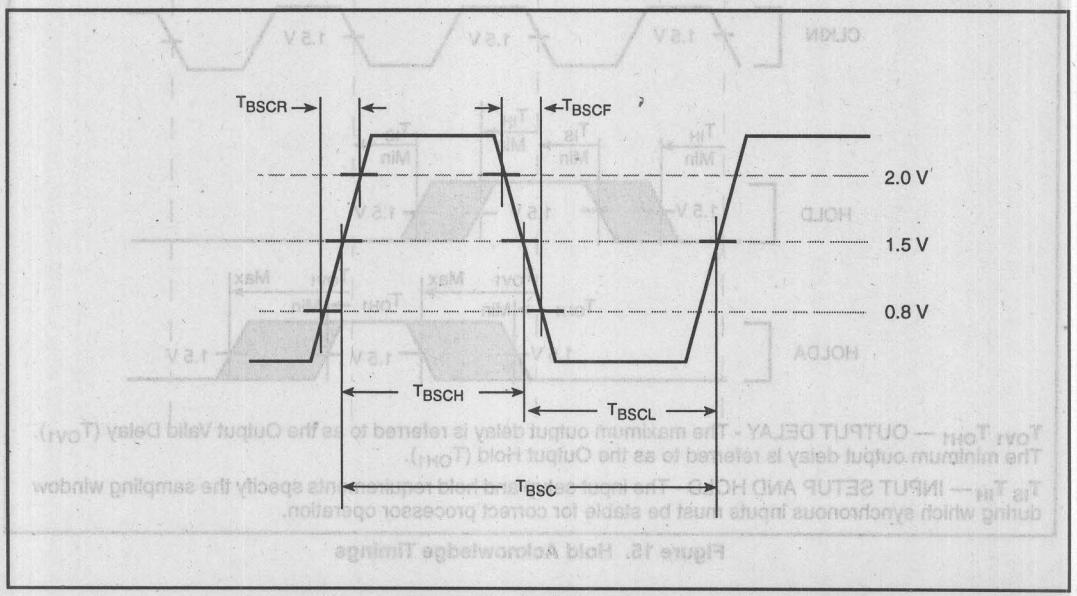


Figure 17. TCK Waveform

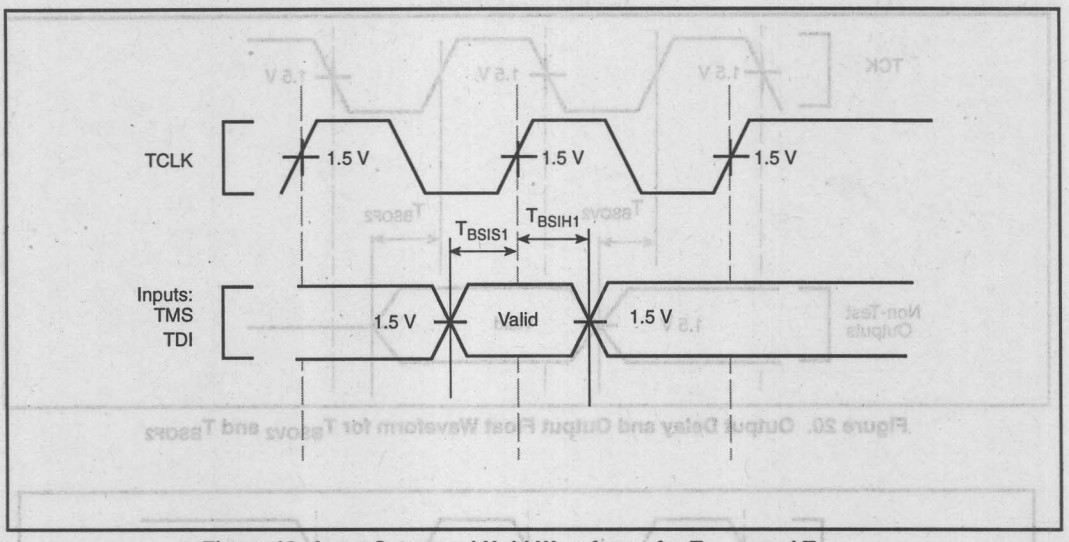


Figure 18. Input Setup and Hold Waveforms for T_{BSIS1} and T_{BSIH1}

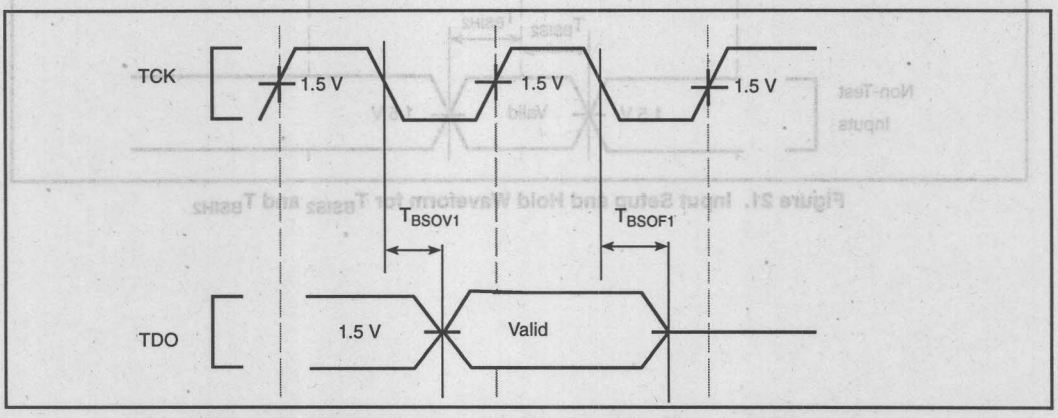


Figure 19. Output Delay and Output Float for T_{BSOV1} and T_{BSOF1}

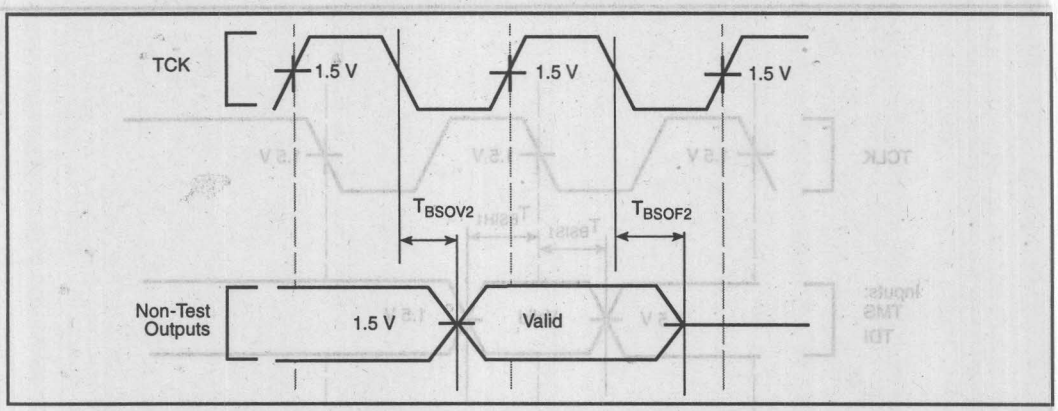


Figure 20. Output Delay and Output Float Waveform for T_{BSOV2} and T_{BSOF2}

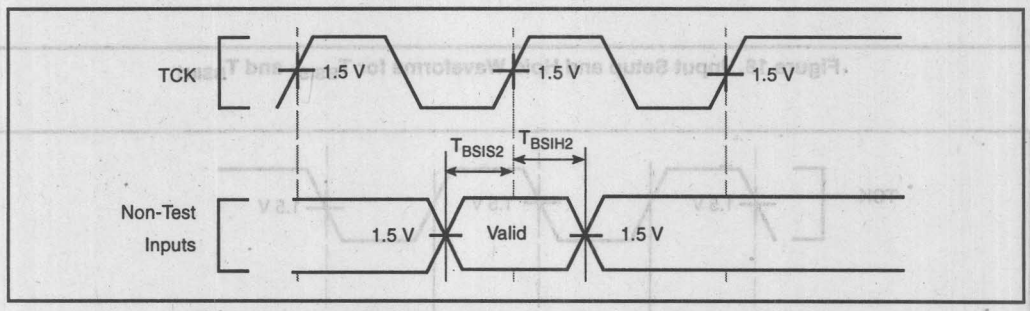


Figure 21. Input Setup and Hold Waveform for T_{BSIS2} and T_{BSIH2}

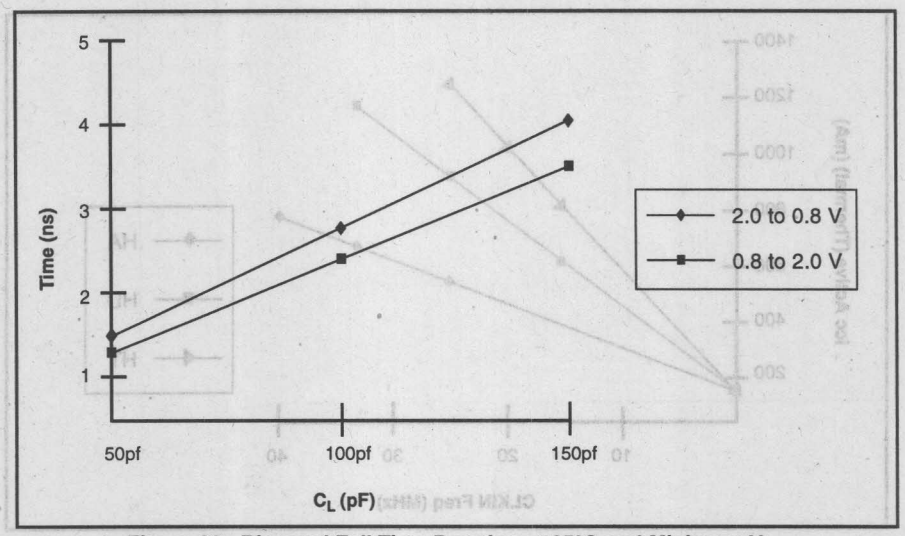


Figure 22. Rise and Fall Time Derating at 85°C and Minimum V_{CC}

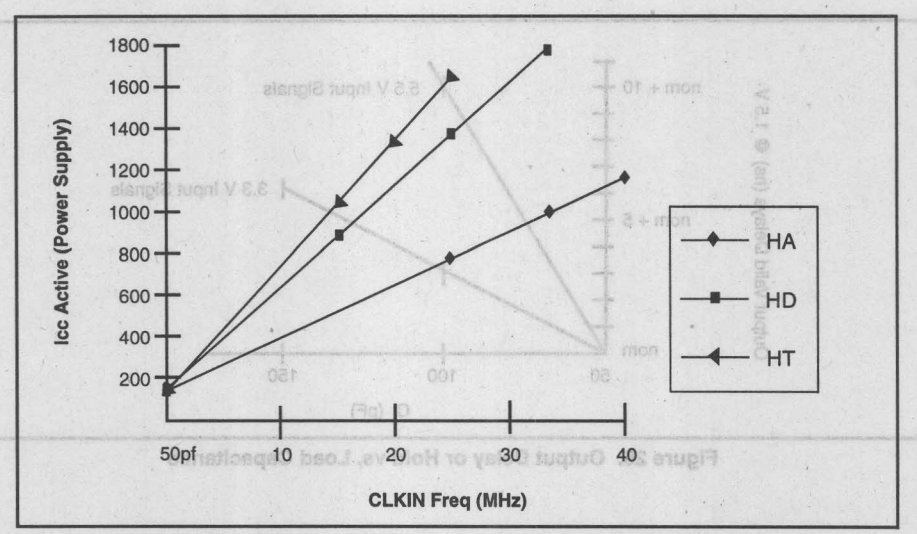


Figure 23. I_{CC} Active (Power Supply) vs. Frequency

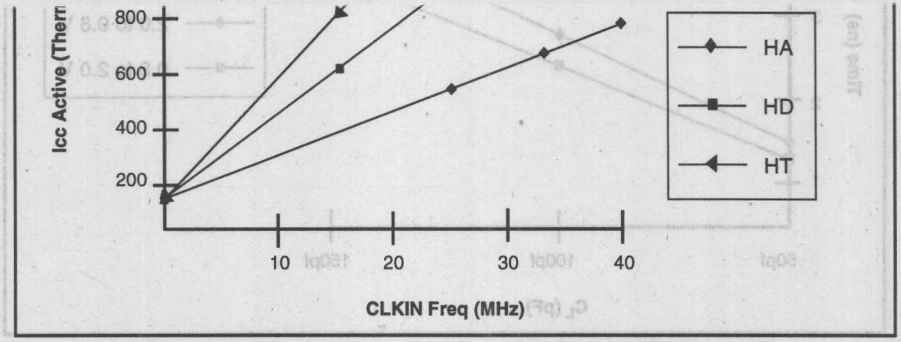


Figure 24. I_{CC} Active (Thermal) vs. Frequency

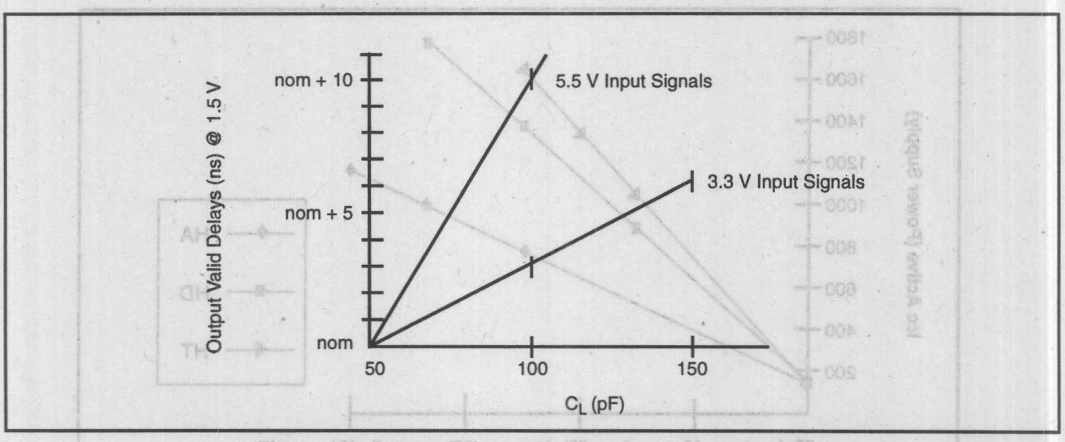


Figure 25. Output Delay or Hold vs. Load Capacitance

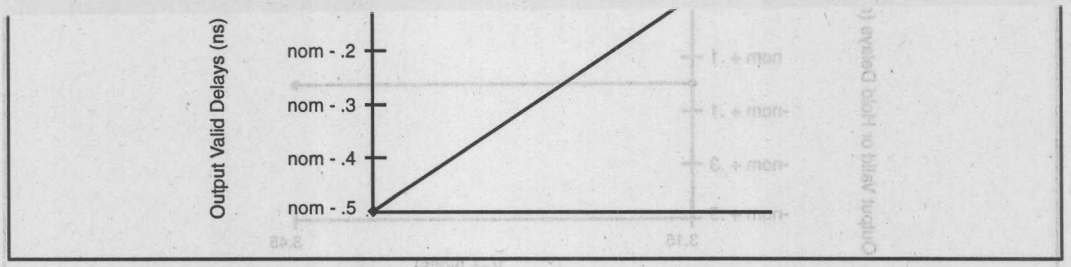


Figure 26. Output Delay vs. Temperature

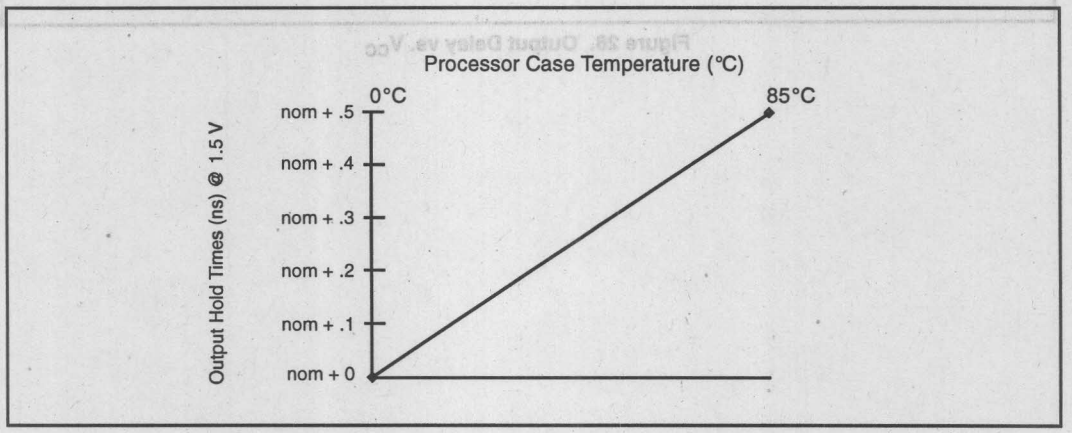


Figure 27. Output Hold Times vs. Temperature

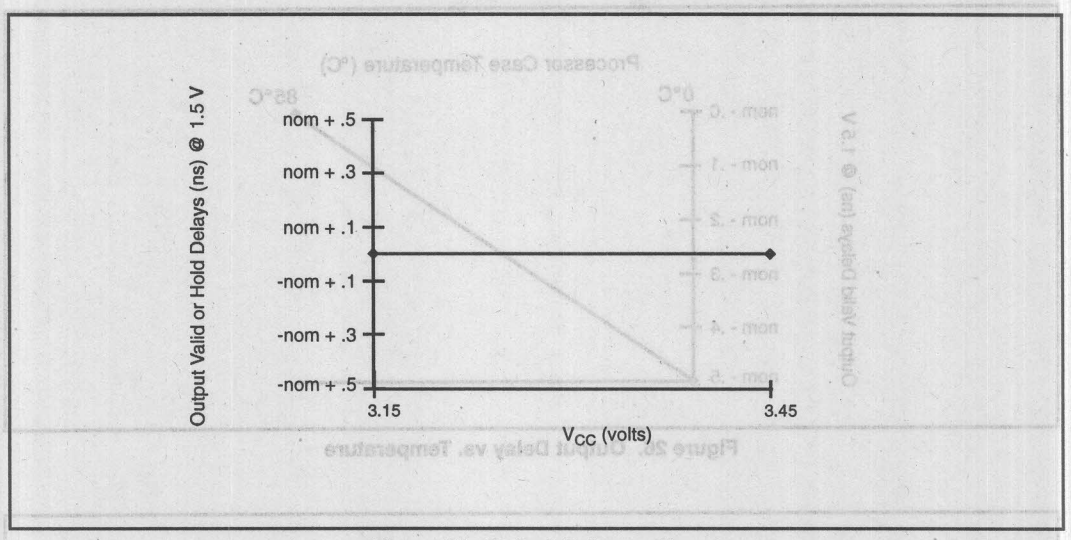


Figure 28. Output Delay vs. V_{CC}

5.0 BUS WAVEFORMS

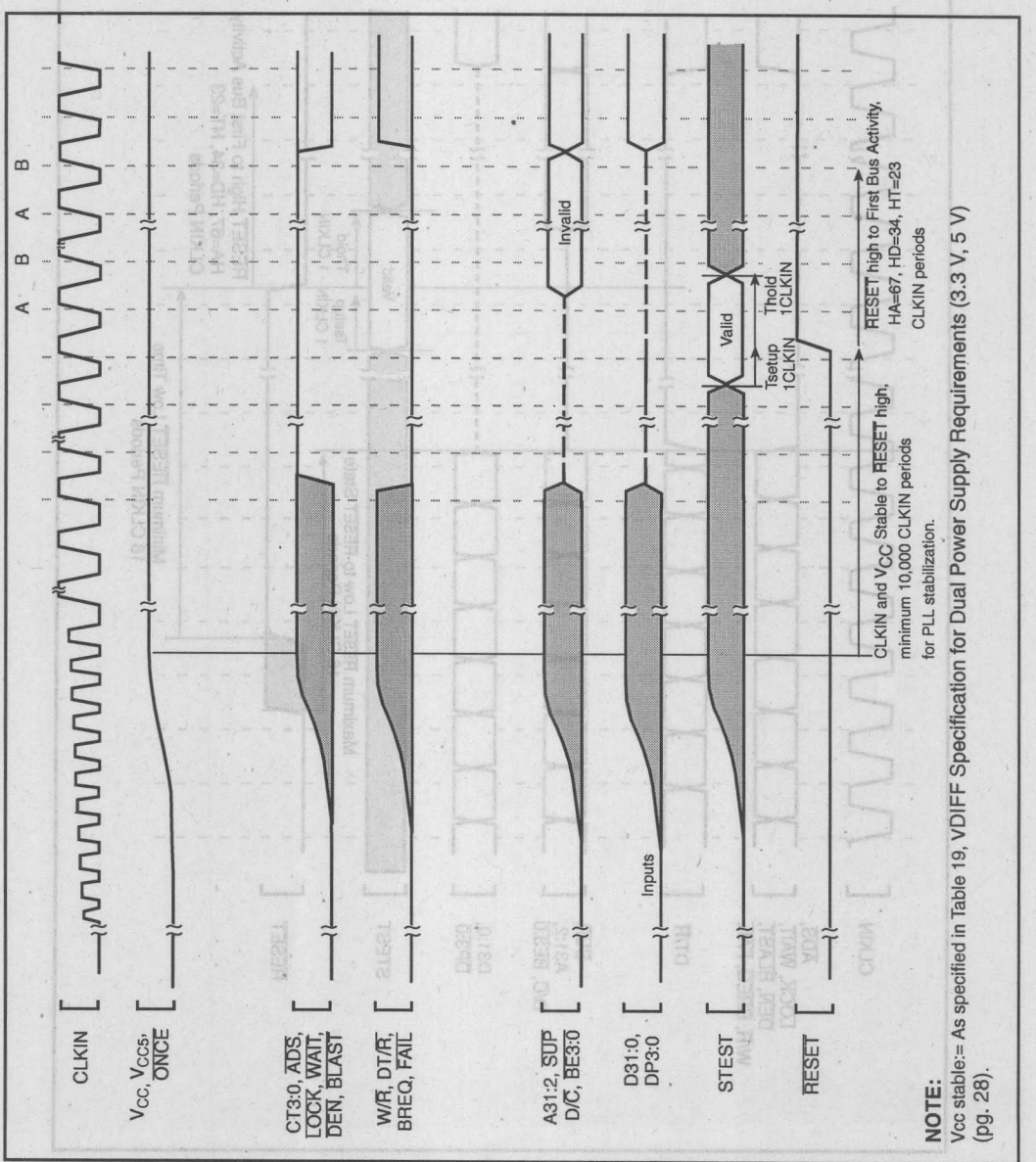


Figure 29. Cold Reset Waveform

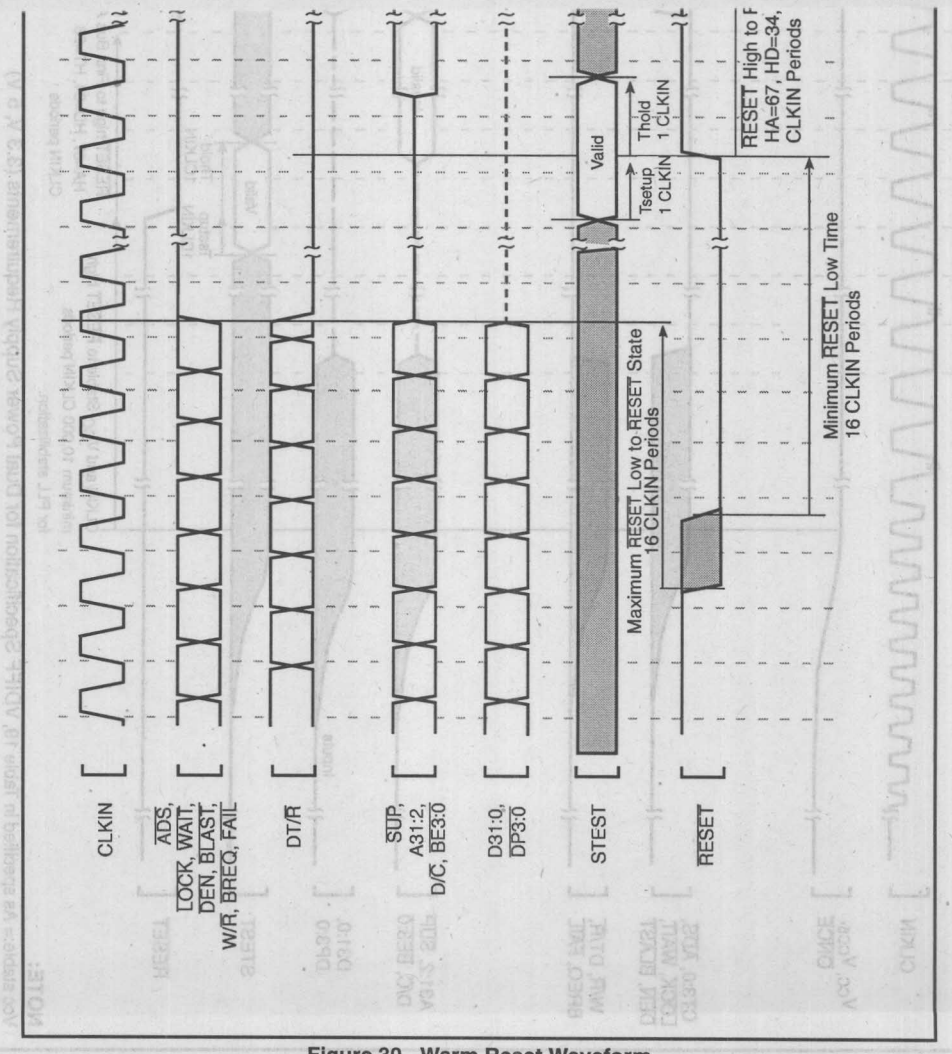


Figure 30. Warm Reset Waveform

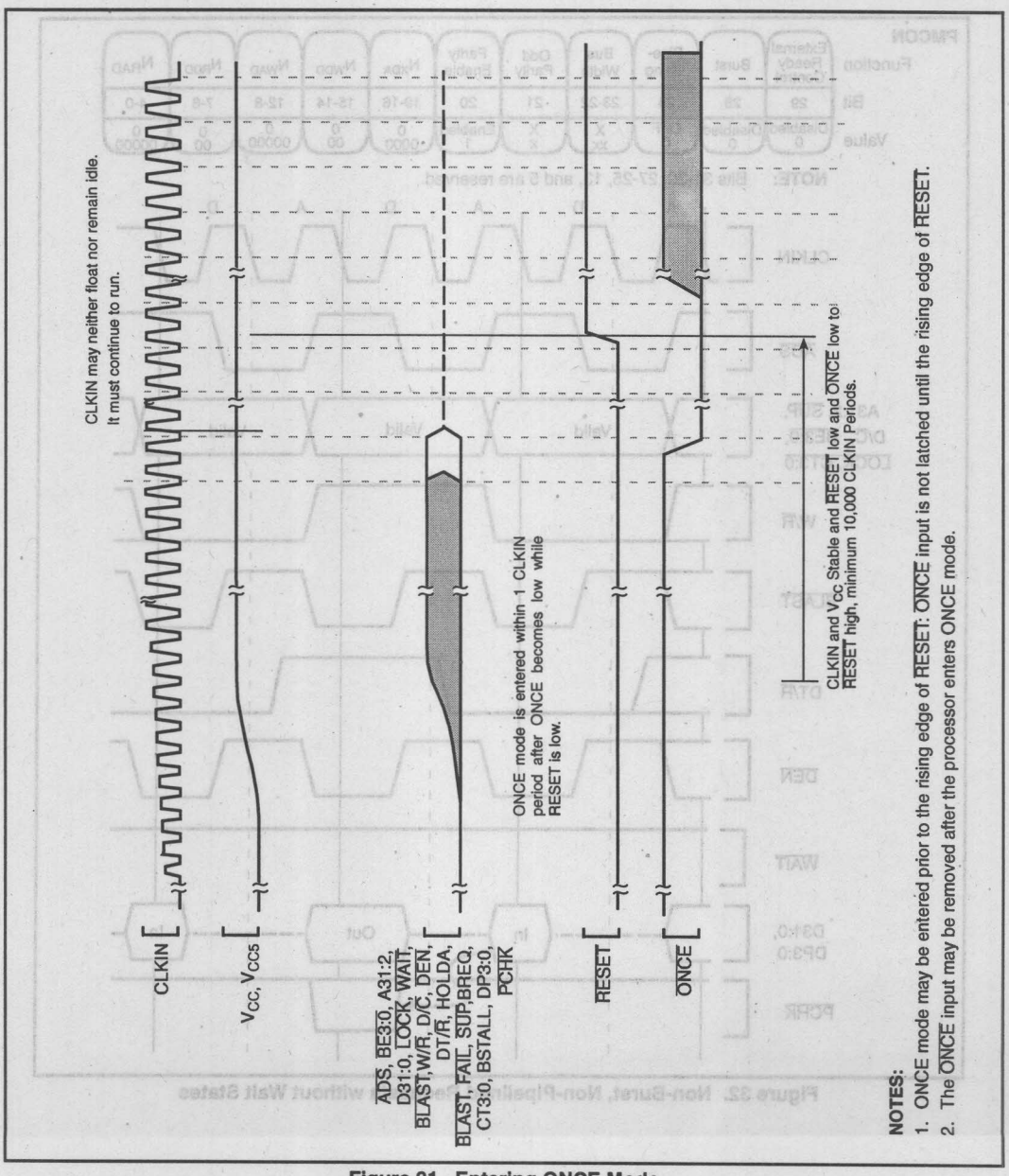


Figure 31. Entering ONCE Mode

PMCON

Function	External Ready Control	Burst	Pipe-Lining	Bus Width	Odd Parity	Parity Enable	N_{XDA}	N_{WDD}	N_{WAD}	N_{RDD}	N_{RAD}
Bit	29	28	24	23-22	21	20	19-16	15-14	12-8	7-6	4-0
Value	Disabled 0	Disabled 0	OFF 0	X xx	X x	Enabled 1	0 0000	0 00	0 00000	0 00	0 00000

NOTE: Bits 31-30, 27-25, 13, and 5 are reserved.

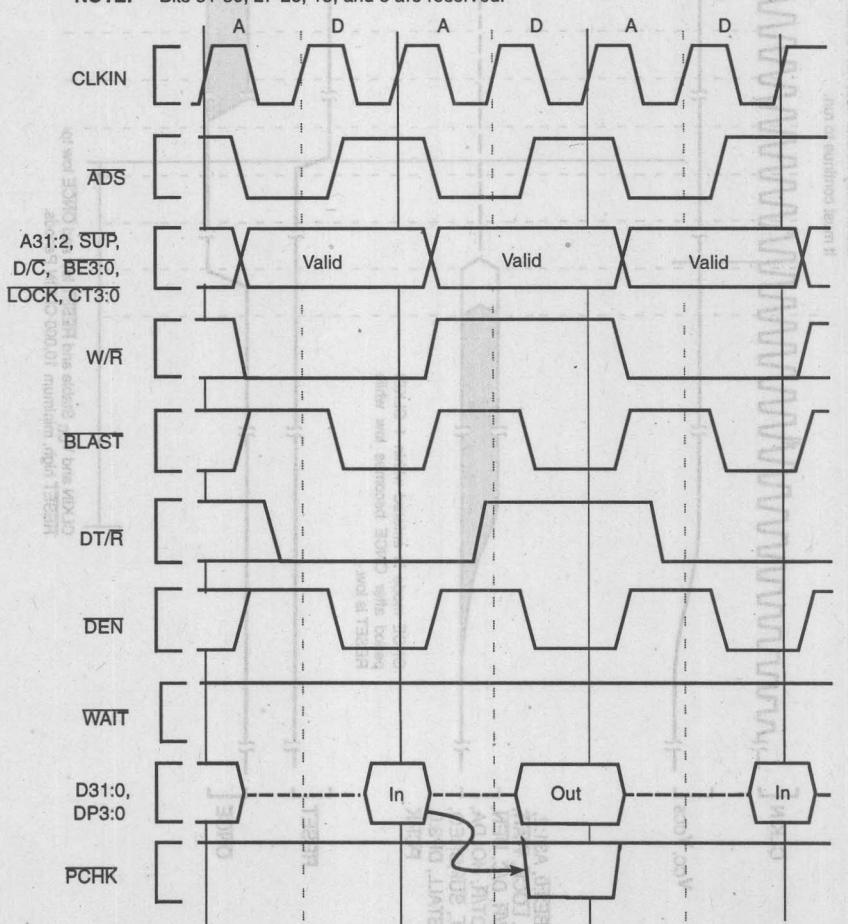


Figure 32. Non-Burst, Non-Pipelined Requests without Wait States

PMCON

Function	External Ready Control	Burst	Pipe-Lining	Bus Width	Odd Parity	Parity Enable	N_{XDA}	N_{WDD}	N_{WAD}	N_{RDD}	N_{RAD}
Bit	29	28	24	23-22	21	20	19-16	15-14	12-8	7-6	4-0
Value	Disabled 0	Disabled 0	OFF 0	X xx	X x	Enabled 1	1 0001	X xx	X xxxxx	X xx	3 00011

NOTE: Bits 31-30, 27-25, 13, and 5 are reserved.

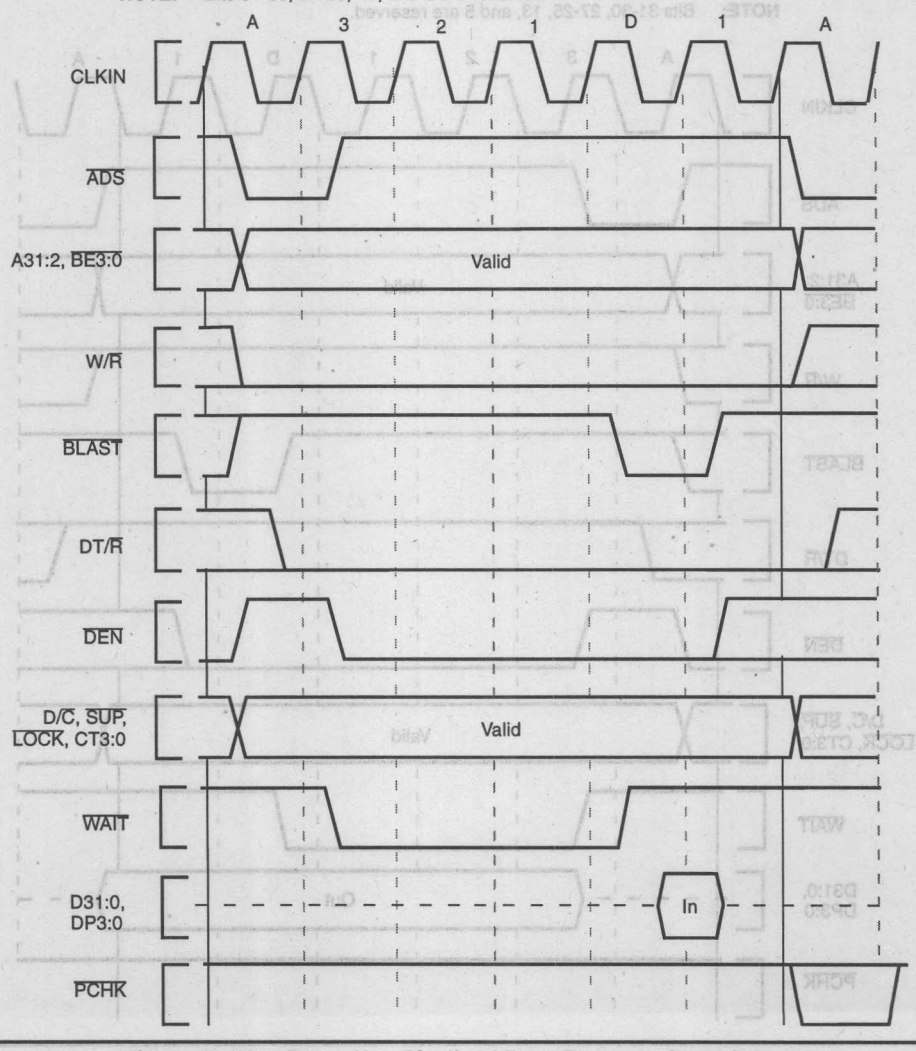


Figure 33. Non-Burst, Non-Pipelined Read Request with Wait States

PMCON

Function	External Ready Control	Burst	Pipe-Lining	Bus Width	Odd Parity	Parity Enable	N_{XDA}	N_{WDD}	N_{WAD}	N_{RDD}	N_{RAD}
Bit	29	28	24	23-22	21	20	19-16	15-14	12-8	7-6	4-0
Value	Disabled 0	Disabled 0	OFF 0	X xx	X x	Enabled 1	1 0001	X xxxxx	3 00011	X xx	X xxxxx

NOTE: Bits 31-30, 27-25, 13, and 5 are reserved.

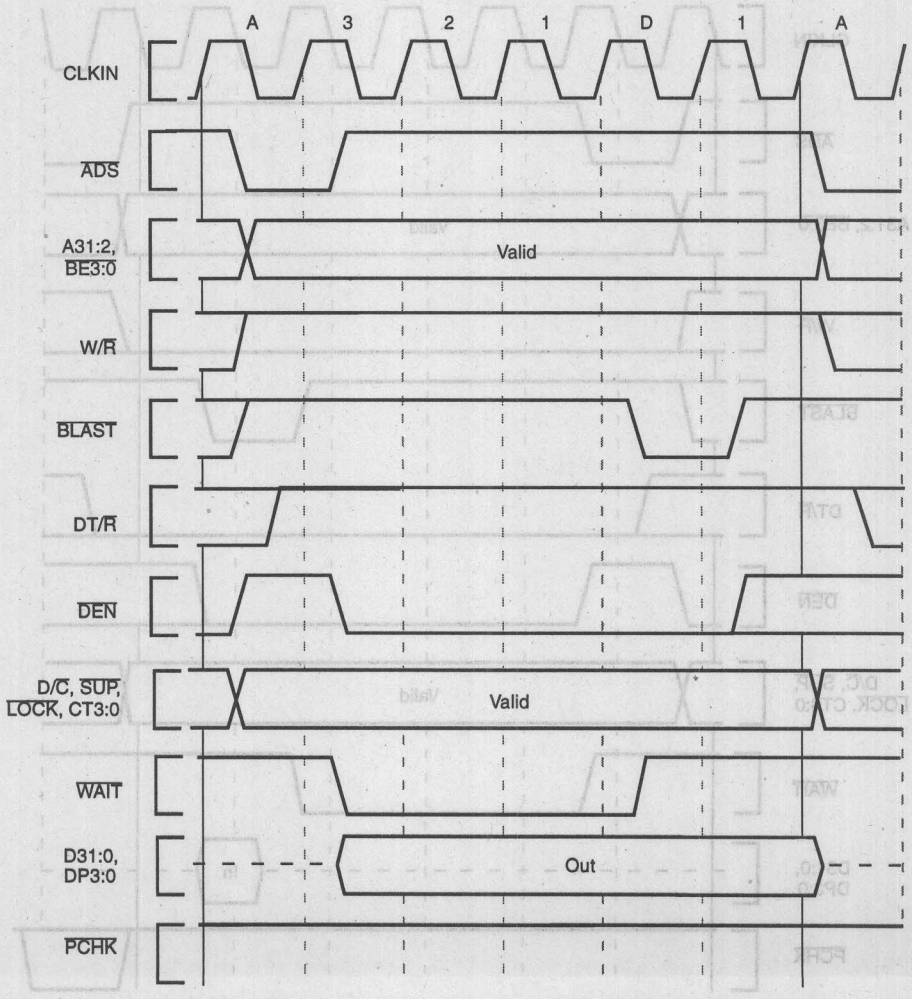


Figure 34. Non-Burst, Non-Pipelined Write Request with Wait States

PMCON

Function	External Ready Control	Burst	Pipe-Lining	Bus Width	Odd Parity	Parity Enable	N_{XDA}	N_{WDD}	N_{WAD}	N_{RDD}	N_{RAD}
Bit	29	28	24	23-22	21	20	19-16	15-14	12-8	7-6	4-0
Value	Disabled 0	Enabled 1	OFF 0	32-Bit 10	X x	Enabled 1	0 0000	X xx	X xxxxx	0 00	0 00000

NOTE: Bits 31-30, 27-25, 13, and 5 are reserved.

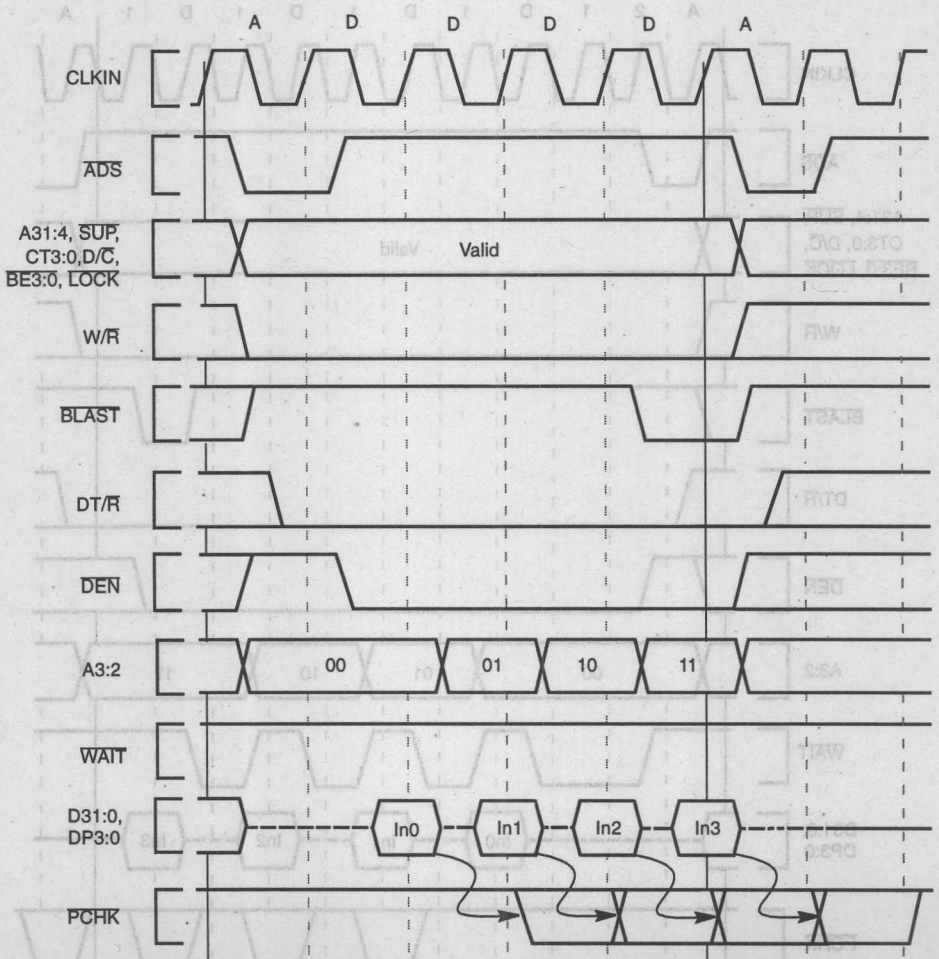


Figure 35. Burst, Non-Pipelined Read Request without Wait States, 32-Bit Bus

PMCON

Function	External Ready Control	Burst	Pipe-Lining	Bus Width	Odd Parity	Parity Enable	N_{XDA}	N_{WDD}	N_{WAD}	N_{RDD}	N_{RAD}
Bit	29	28	24	23-22	21	20	19-16	15-14	12-8	7-6	4-0
Value	Disabled 0	Enabled 1	OFF 0	32-Bit 10	X x	Enabled 1	1 0001	X xx	X xxxxx	1 01	2 00010

NOTE: Bits 31-30, 27-25, 13, and 5 are reserved.

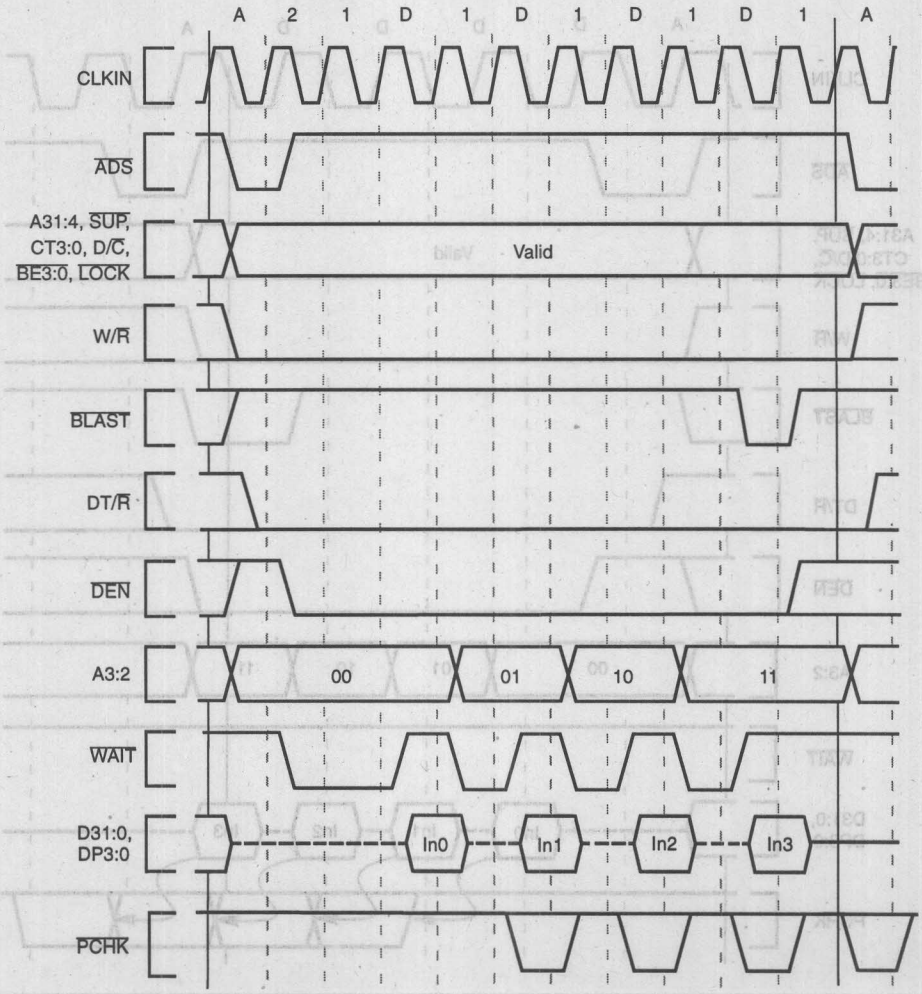


Figure 36. Burst, Non-Pipelined Read Request with Wait States, 32-Bit Bus

PMCON Function	External Ready Control	Burst	Pipe-Lining	Bus Width	Odd Parity	Parity Enable	N_{XDA}	N_{WDD}	N_{WAD}	N_{RDD}	N_{RAD}
Bit	29	28	24	23-22	21	20	19-16	15-14	12-8	7-6	4-0
Value	Disabled 0	Enabled 1	OFF 0	32-Bit 10	X x	Enabled 1	0 0000	0 00	0 00000	X xx	X xxxxx

NOTE: Bits 31-30, 27-25, 13, and 5 are reserved.

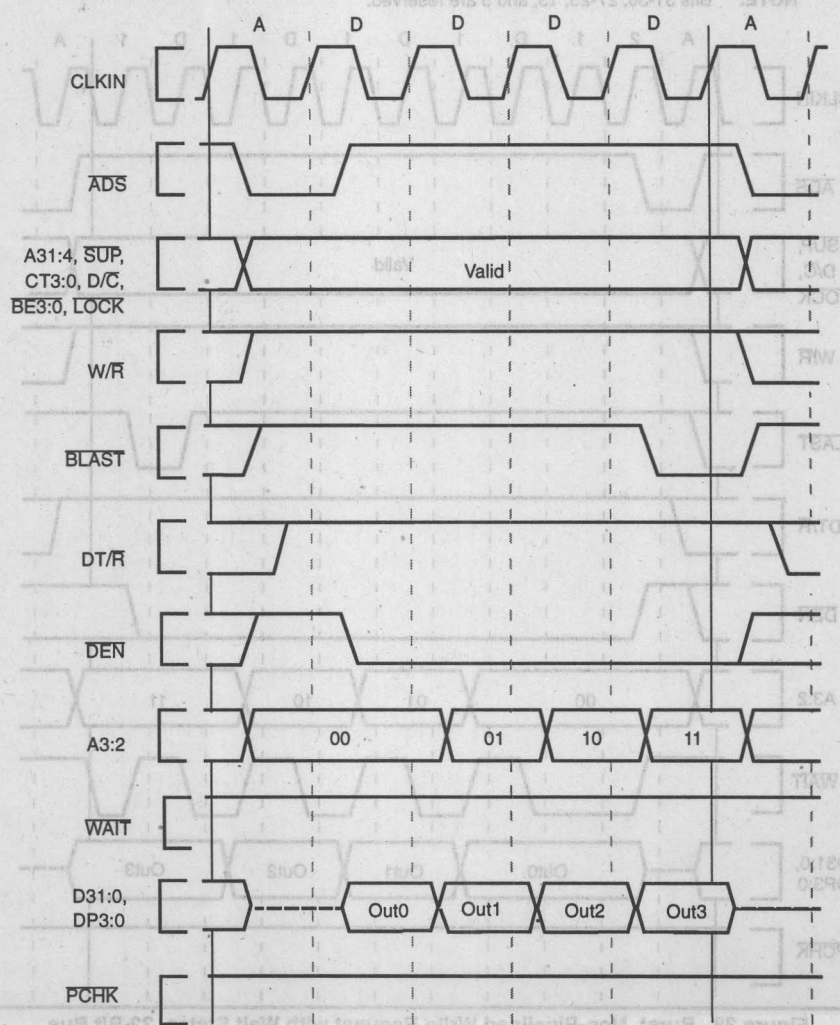


Figure 37. Burst, Non-Pipelined Write Request without Wait States, 32-Bit Bus

PMCON

Function	External Ready Control	Burst	Pipe-Lining	Bus Width	Odd Parity	Parity Enable	N_{XDA}	N_{WDD}	N_{WAD}	N_{RDD}	N_{RAD}
Bit	29	28	24	23-22	21	20	19-16	15-14	12-8	7-6	4-0
Value	Disabled 0	Enabled 1	OFF 0	32-bit 10	X x	Enabled 1	1 0001	1 01	2 00010	X xx	X xxxxx

NOTE: Bits 31-30, 27-25, 13, and 5 are reserved.

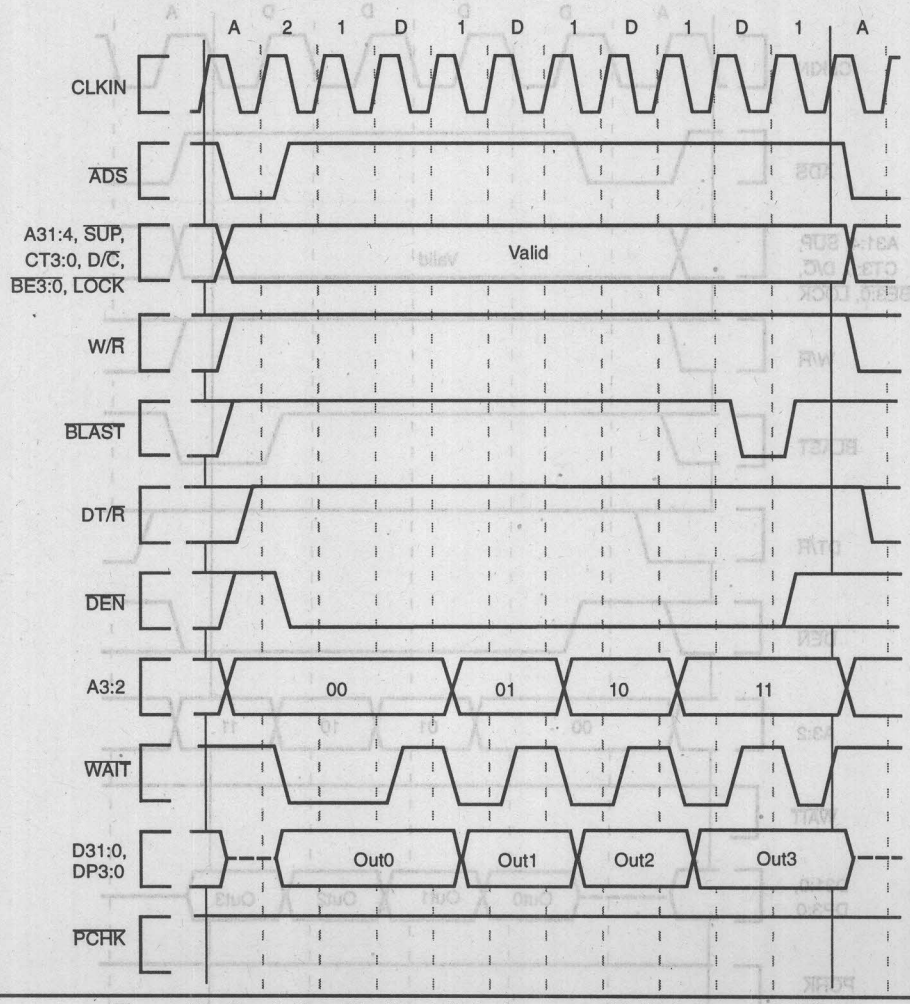


Figure 38. Burst, Non-Pipelined Write Request with Wait States, 32-Bit Bus

PMCON

Function	External Ready Control	Burst	Pipe-Lining	Bus Width	Odd Parity	Parity Enable	N_{XDA}	N_{WDD}	N_{WAD}	N_{RDD}	N_{RAD}
Bit	29	28	24	23-22	21	20	19-16	15-14	12-8	7-6	4-0
Value	Disabled 0	Enabled 1	OFF 0	16-Bit 01	X x	Enabled 1	1 0001	X xx	X xxxxx	1 01	2 00010

NOTE: Bits 31-30, 27-25, 13, and 5 are reserved.

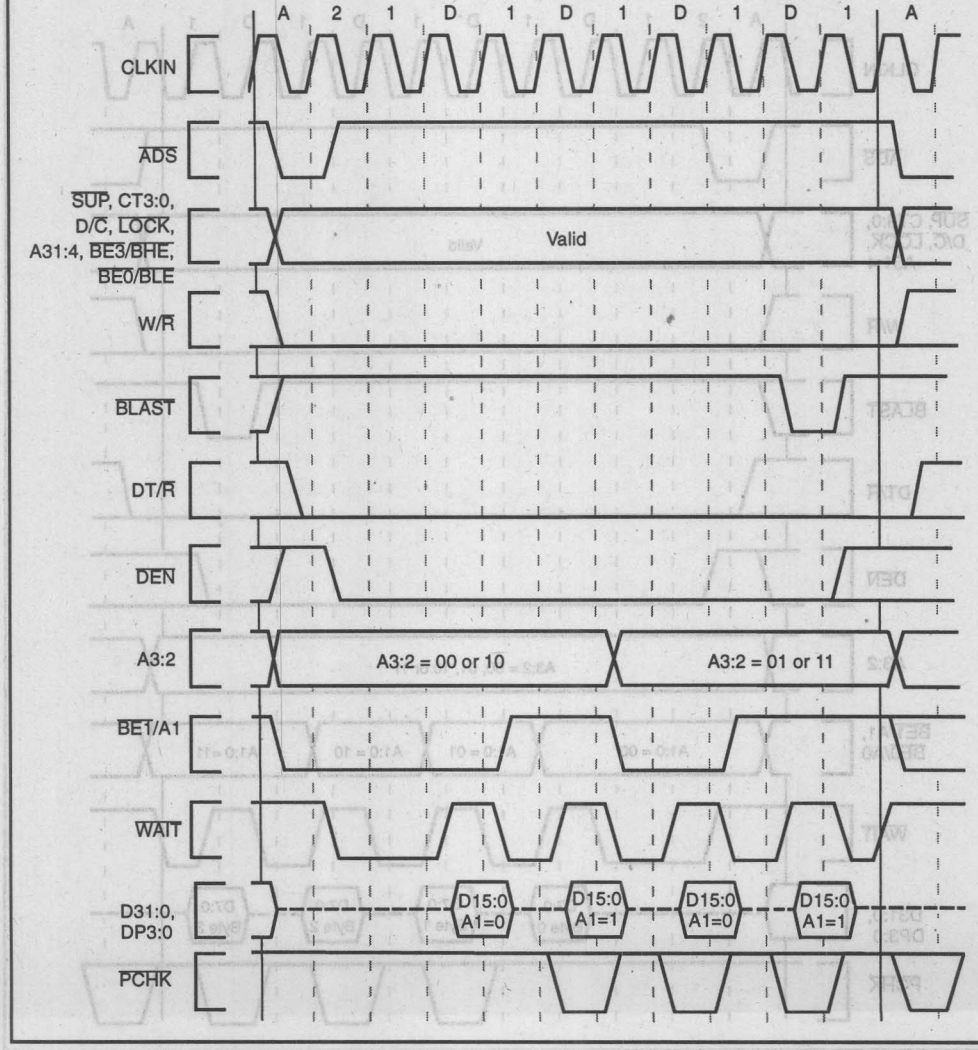


Figure 39. Burst, Non-Pipelined Read Request with Wait States, 16-Bit Bus

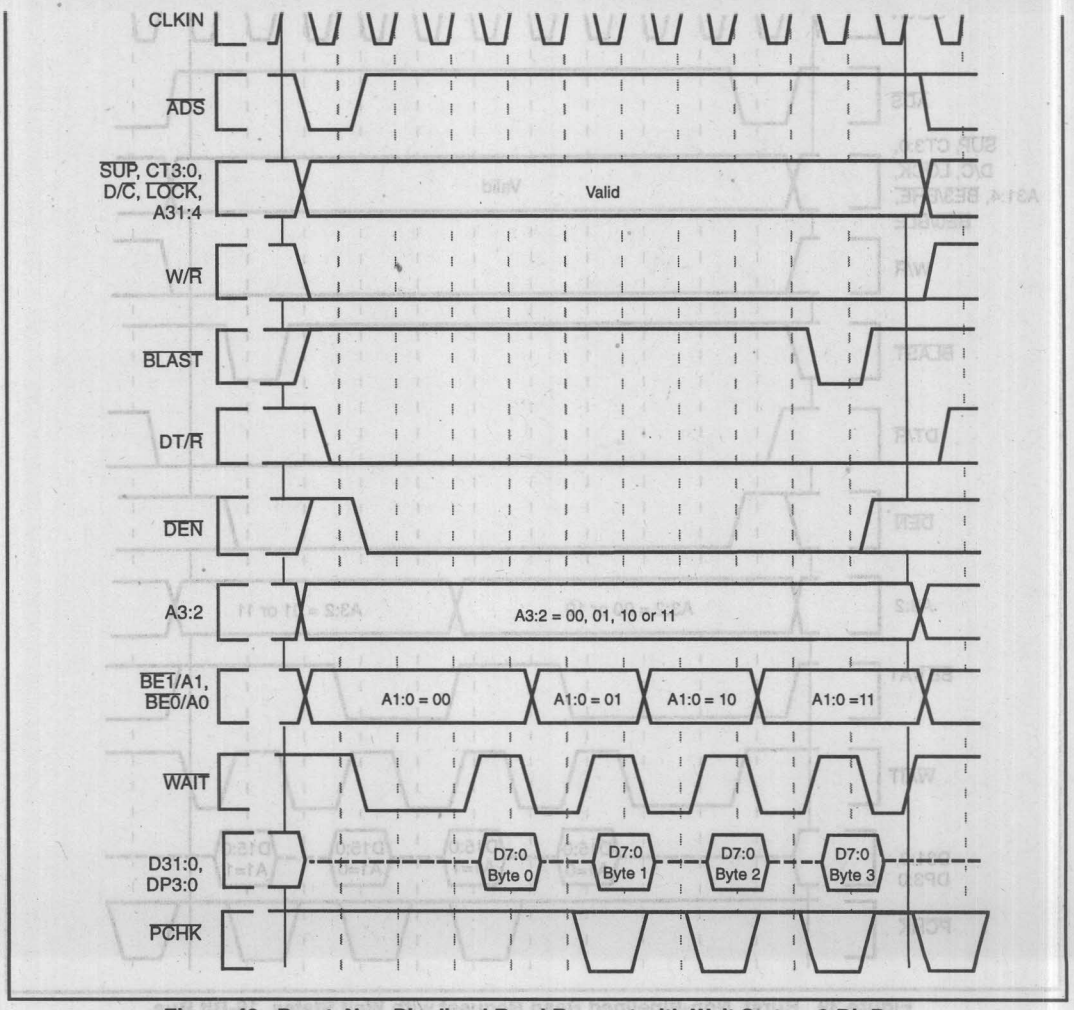
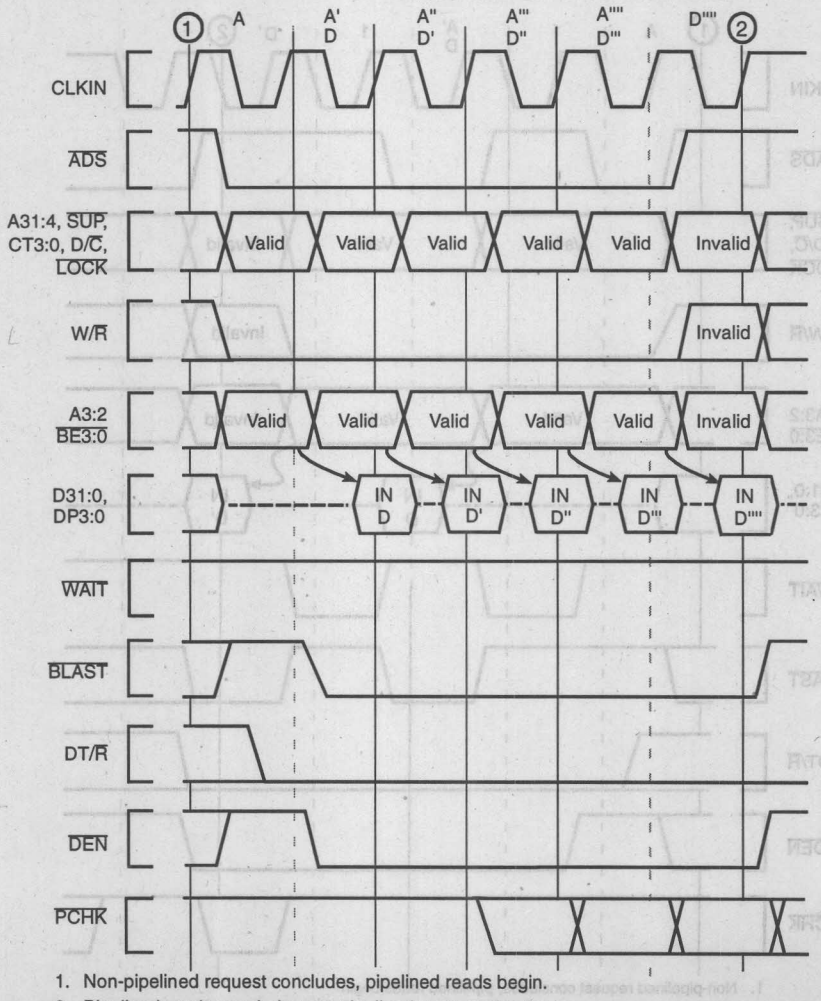


Figure 40. Burst, Non-Pipelined Read Request with Wait States, 8-Bit Bus

PMCON

Function	External Ready Control	Burst	Pipe-Lining	Bus Width	Odd Parity	Parity Enable	N_{XDA}	N_{WDD}	N_{WAD}	N_{RDD}	N_{RAD}
Bit	29	28	24	23-22	21	20	19-16	15-14	12-8	7-6	4-0
Value	X x	Disabled 0	ON 1	32-Bit 10	X x	Enabled 1	X xxxx	X xx	X xxxxx	X xx	0 00000

NOTE: Bits 31-30, 27-25, 13, and 5 are reserved.



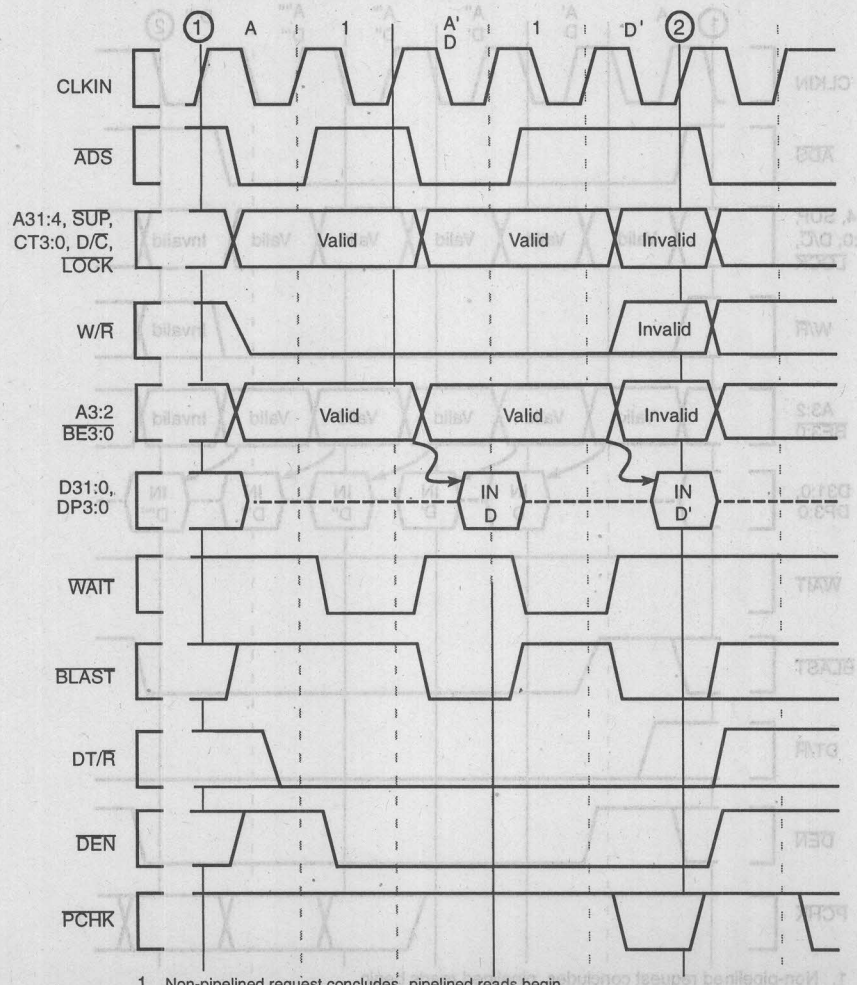
1. Non-pipelined request concludes, pipelined reads begin.
2. Pipelined reads conclude, non-pipelined requests begin.

Figure 41. Non-Burst, Pipelined Read Request without Wait States, 32-Bit Bus

PMCON

Function	External Ready Control	Burst	Pipe-Lining	Bus Width	Odd Parity	Parity Enable	N_{XDA}	N_{WDD}	N_{WAD}	N_{RDD}	N_{RAD}
Bit	29	28	24	23-22	21	20	19-16	15-14	12-8	7-6	4-0
Value	X x	Disabled 0	ON 1	32-Bit 10	X x	Enabled 1	X xxxx	X xx	X xxxxx	X xx	1 00001

NOTE: Bits 31-30, 27-25, 13, and 5 are reserved.

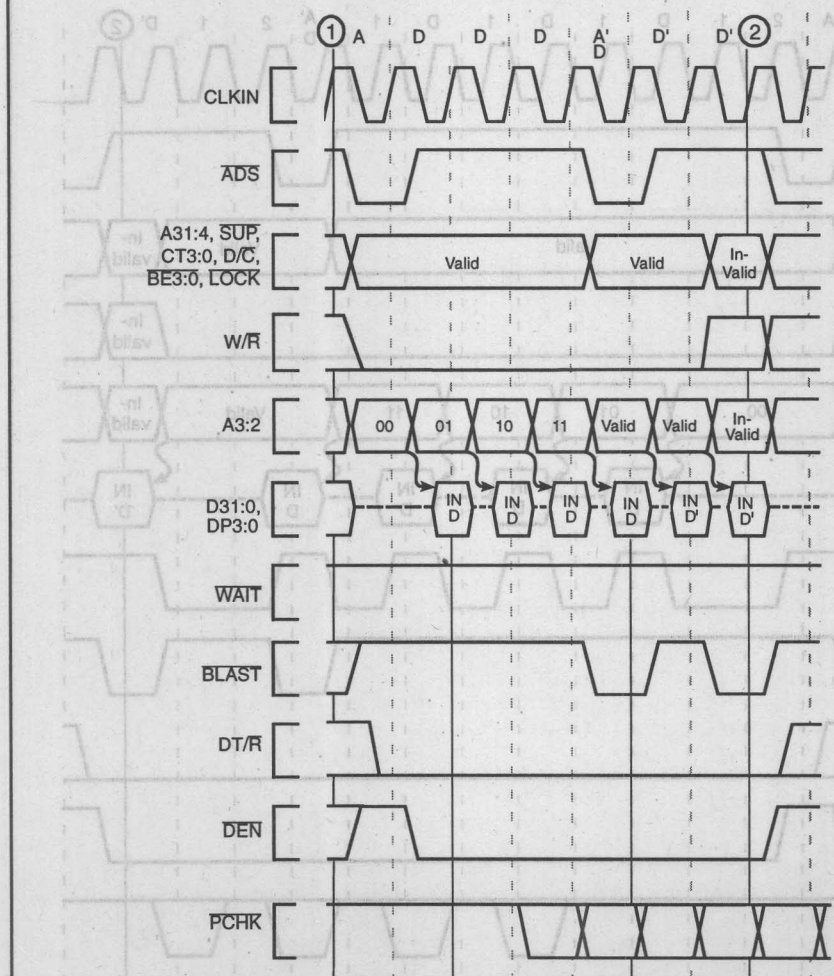


1. Non-pipelined request concludes, pipelined reads begin
2. Pipelined reads conclude, non-pipelined requests begin

Figure 42. Non-Burst, Pipelined Read Request with Wait States, 32-Bit Bus

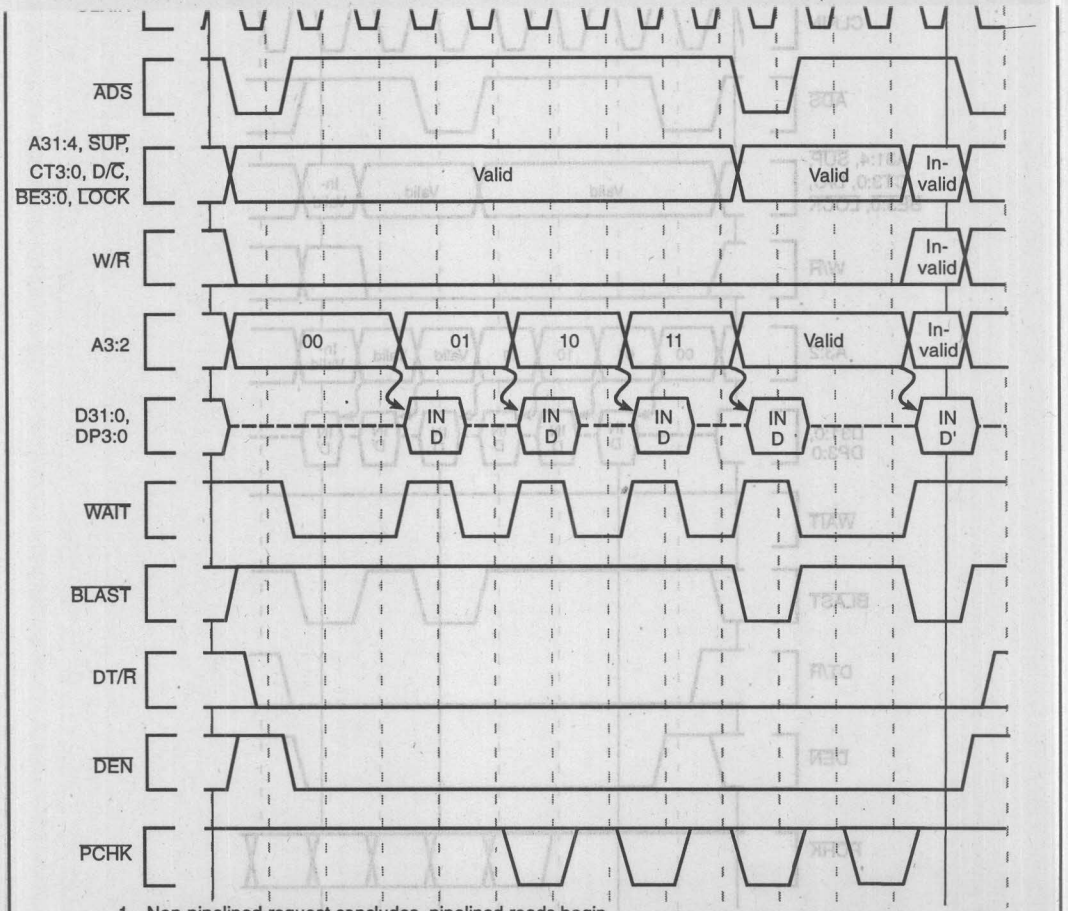
PMCON Function	External Ready Control	Burst	Pipe-Lining	Bus Width	Odd Parity	Parity Enable	N_{XDA}	N_{WDD}	N_{WAD}	N_{RDD}	N_{RAD}
Bit	29	28	24	23-22	21	20	19-16	15-14	12-8	7-6	4-0
Value	X x	Enabled 1	ON 1	32-Bit 10	X x	Enabled 1	X xxxx	X xx	X xxxxx	0 00	0 00000

NOTE: Bits 31-30, 27-25, 13, and 5 are reserved.



1. Non-pipelined request concludes, pipelined reads begin
2. Pipelined reads conclude, non-pipelined requests begin

Figure 43. Burst, Pipelined Read Request without Wait States, 32-Bit Bus



1. Non-pipelined request concludes, pipelined reads begin.
2. Pipelined reads conclude, non-pipelined requests begin.

Figure 44. Burst, Pipelined Read Request with Wait States, 32-Bit Bus

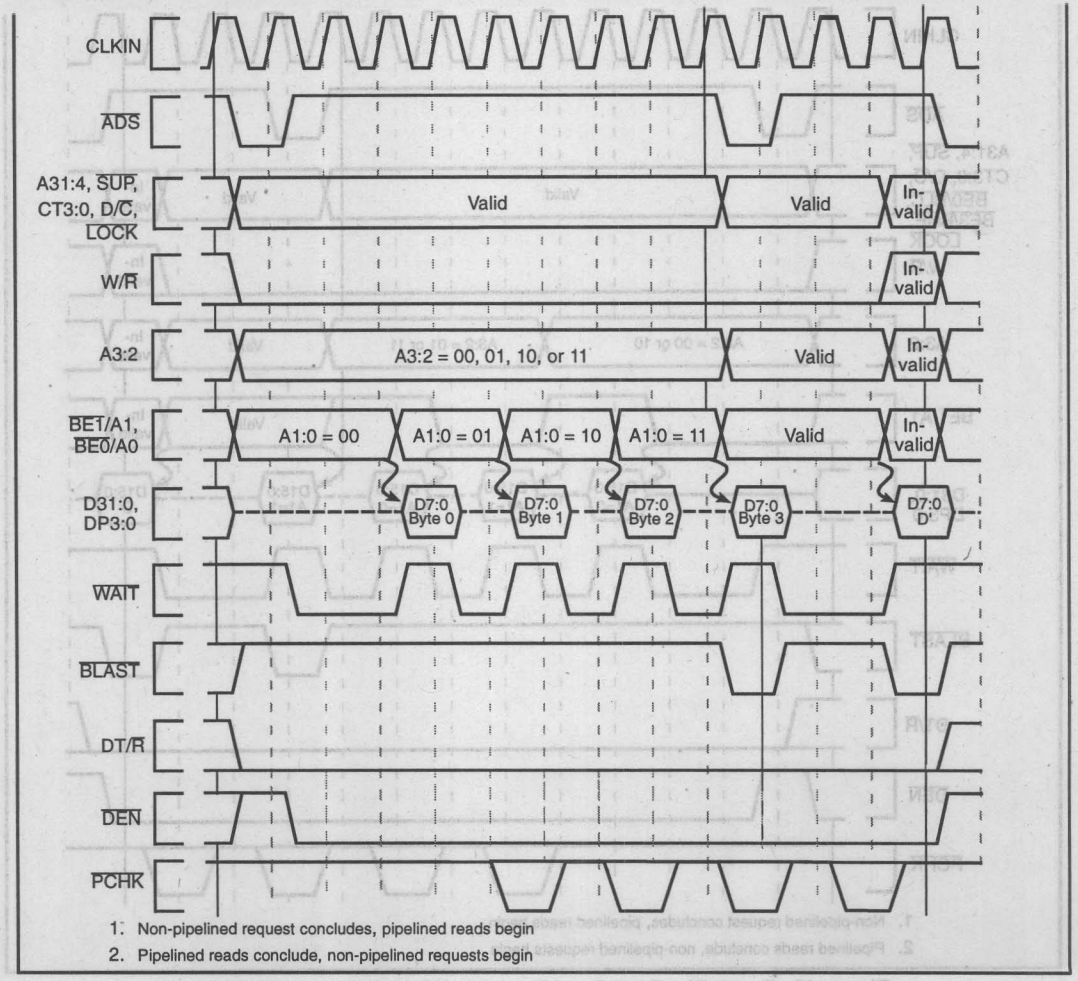
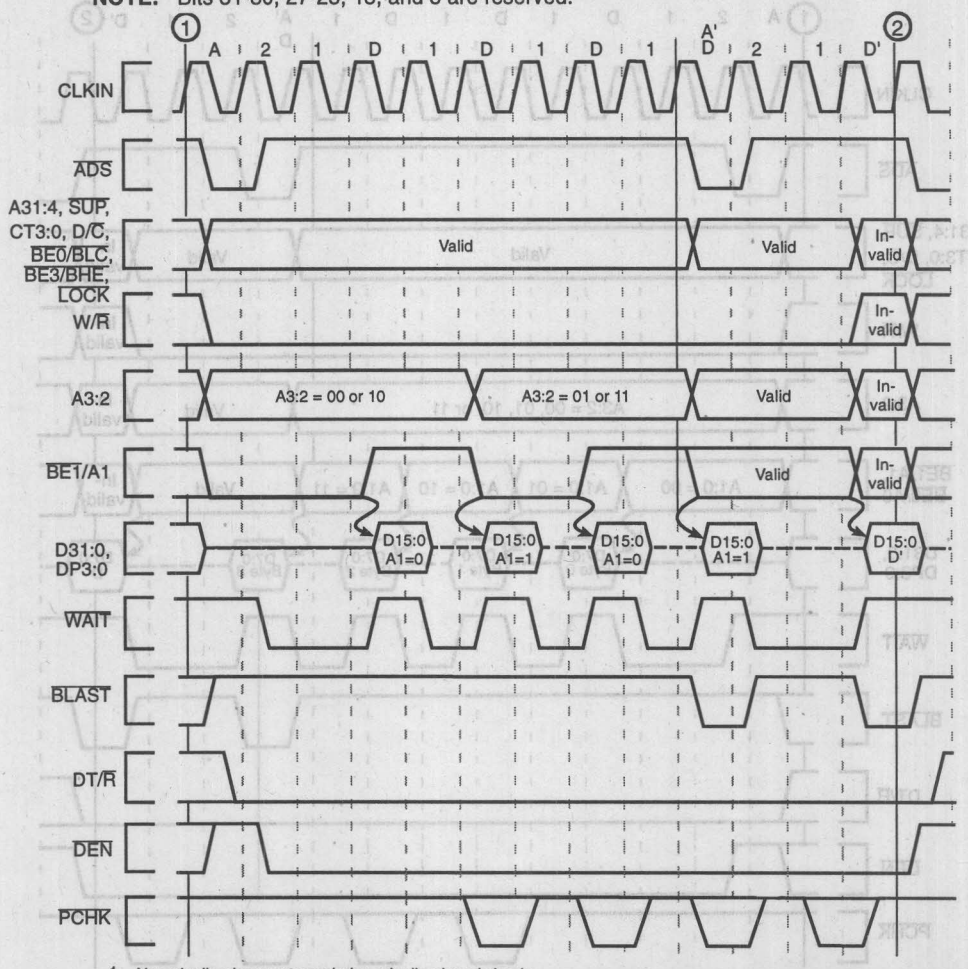


Figure 45. Burst, Pipelined Read Request with Wait States, 8-Bit Bus

PMCON

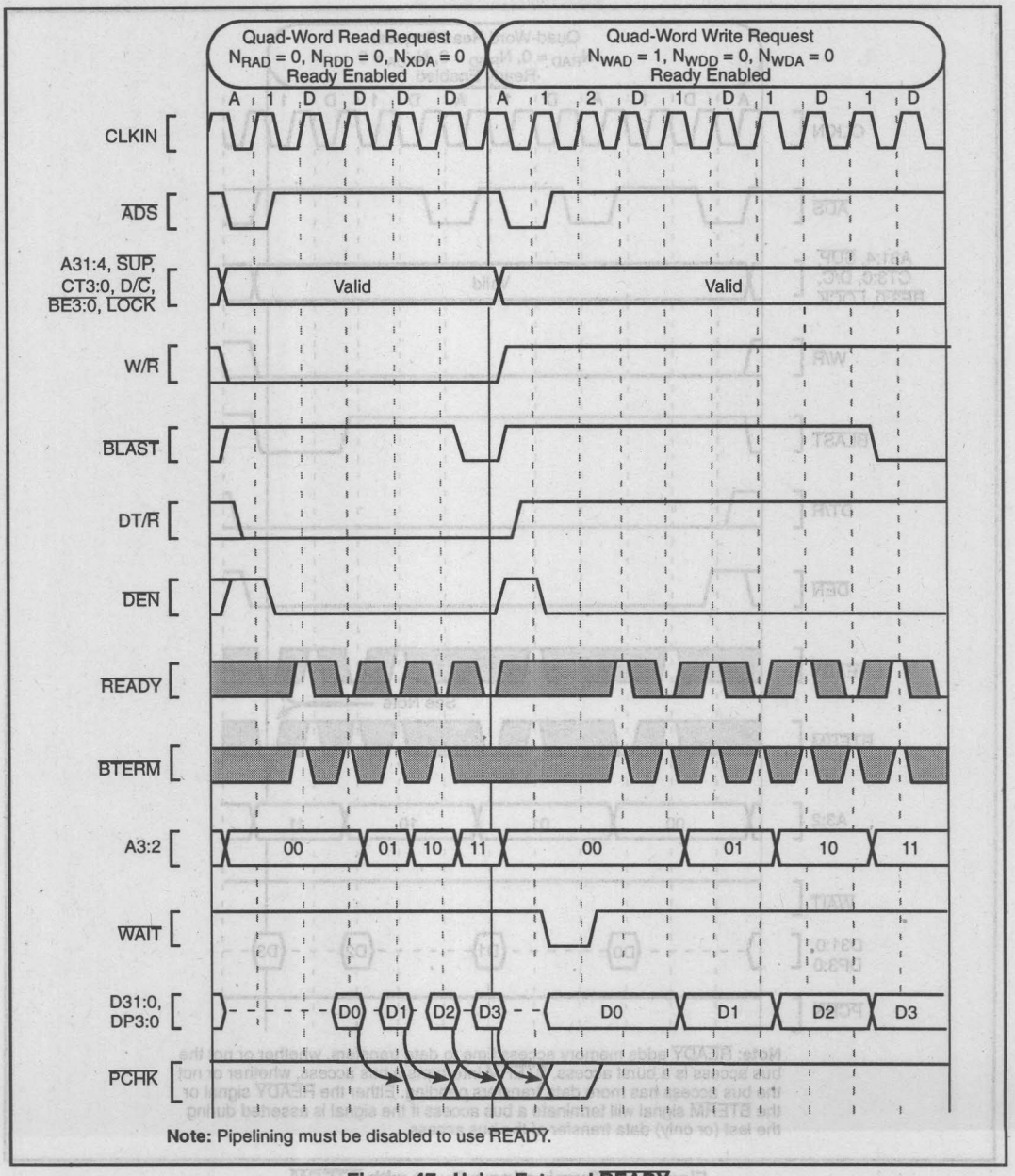
Function	External Ready Control	Burst	Pipe-Lining	Bus Width	Odd Parity	Parity Enable	N_{XDA}	N_{WDD}	N_{WAD}	N_{RDD}	N_{RAD}
Bit	29	28	24	23-22	21	20	19-16	15-14	12-8	7-6	4-0
Value	X x	Enabled 1	ON 1	16-Bit 01	X x	Enabled 1	X xxxx	X xx	X xxxxx	1 01	2 00010

NOTE: Bits 31-30, 27-25, 13, and 5 are reserved.



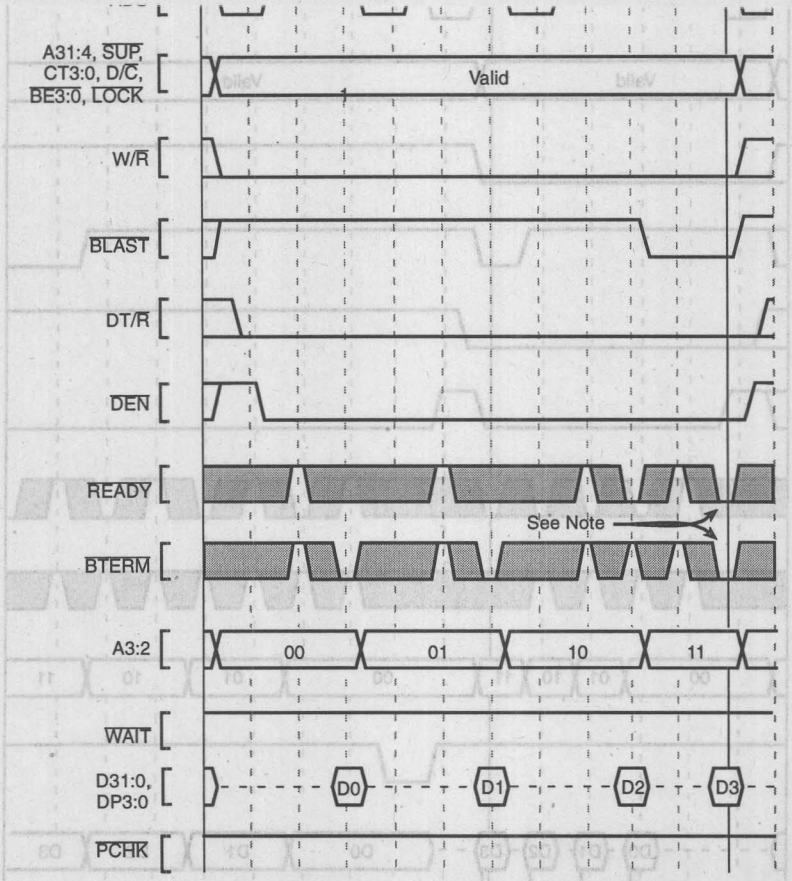
1. Non-pipelined request concludes, pipelined reads begin
2. Pipelined reads conclude, non-pipelined requests begin

Figure 46. Burst, Pipelined Read Request with Wait States, 16-Bit Bus



Note: Pipelining must be disabled to use READY.

Figure 47. Using External READY



Note: READY adds memory access time to data transfers, whether or not the bus access is a burst access. BTERM interrupts a bus access, whether or not the bus access has more data transfers pending. Either the READY signal or the BTERM signal will terminate a bus access if the signal is asserted during the last (or only) data transfer of the bus access.

Figure 48. Terminating a Burst with BTERM

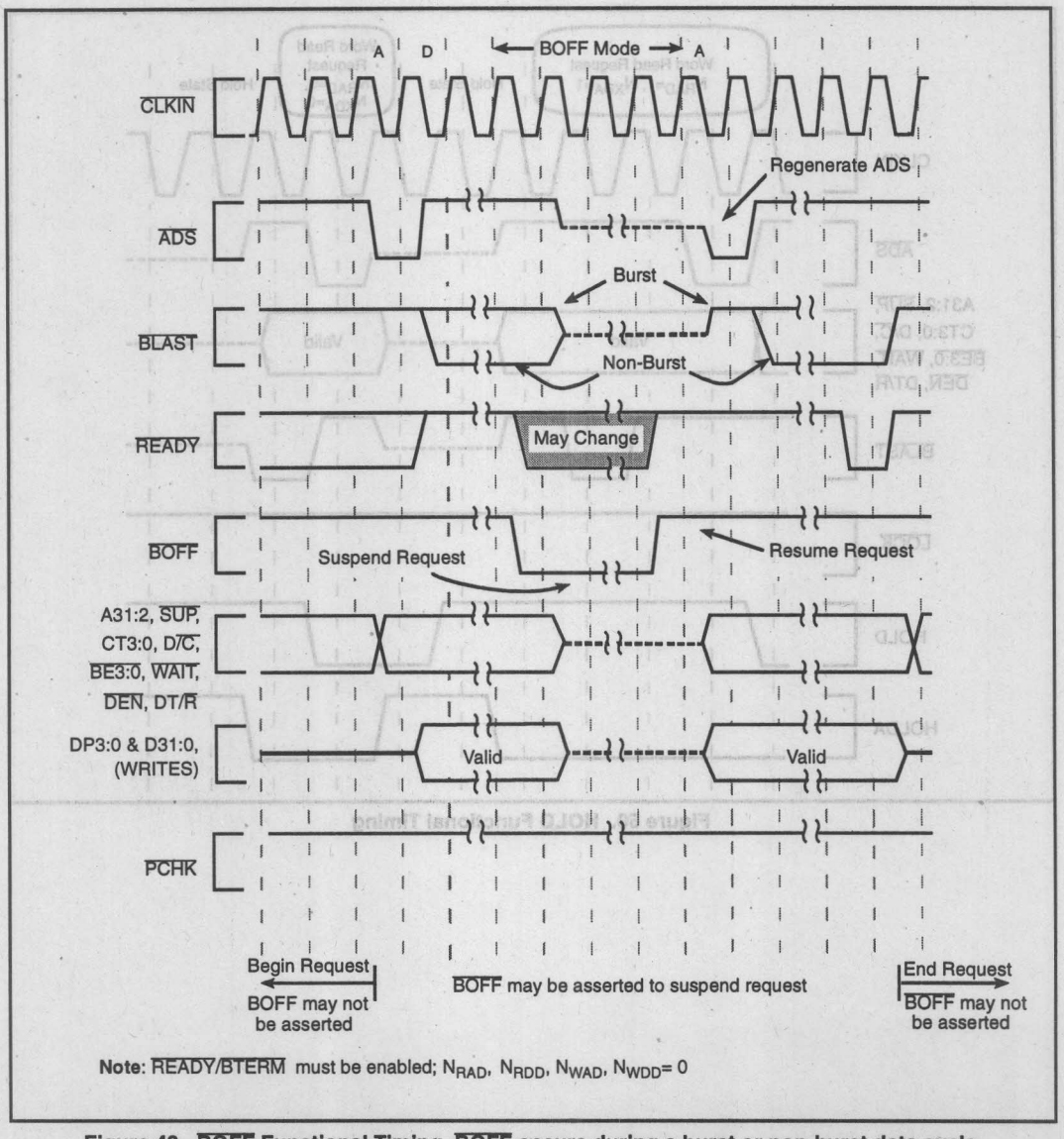


Figure 49. BOFF Functional Timing. BOFF occurs during a burst or non-burst data cycle.

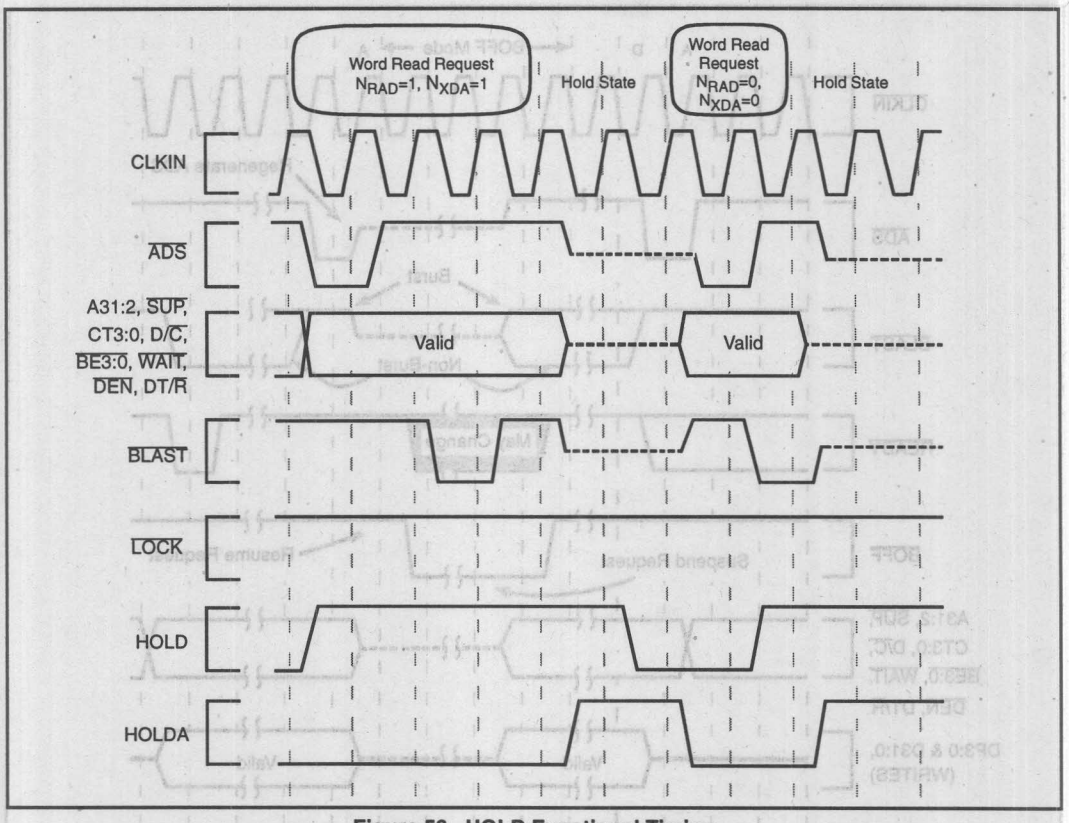


Figure 50. HOLD Functional Timing

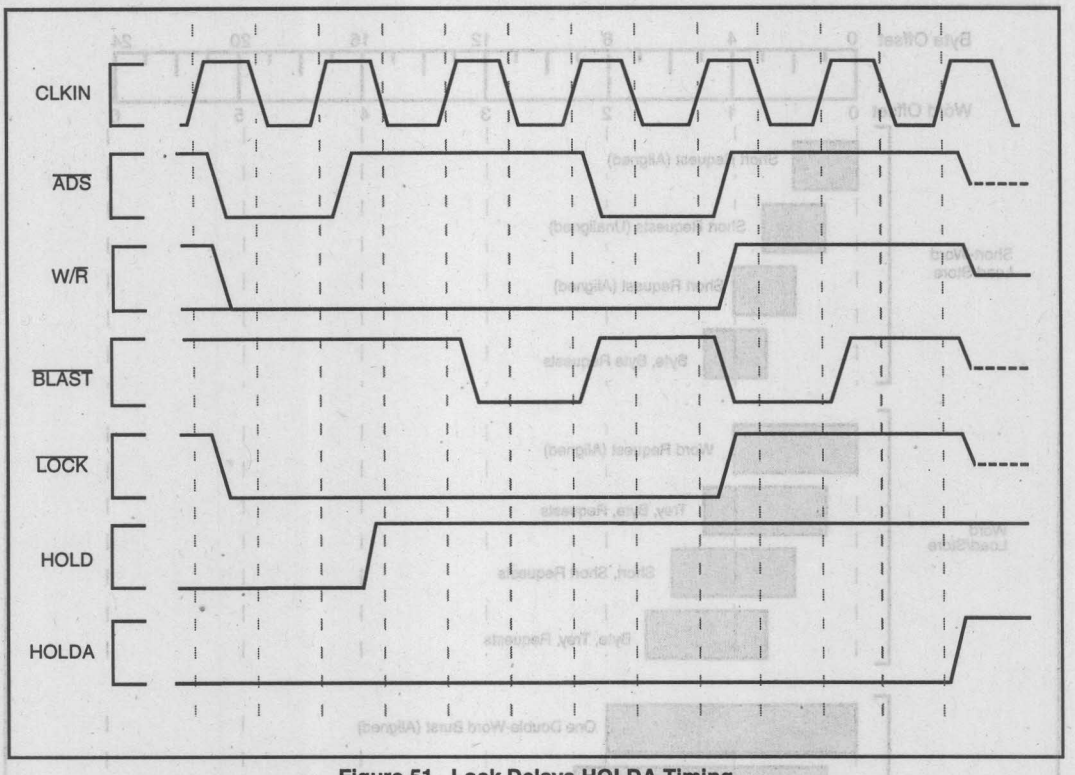


Figure 51. Lock Delays HOLDA Timing

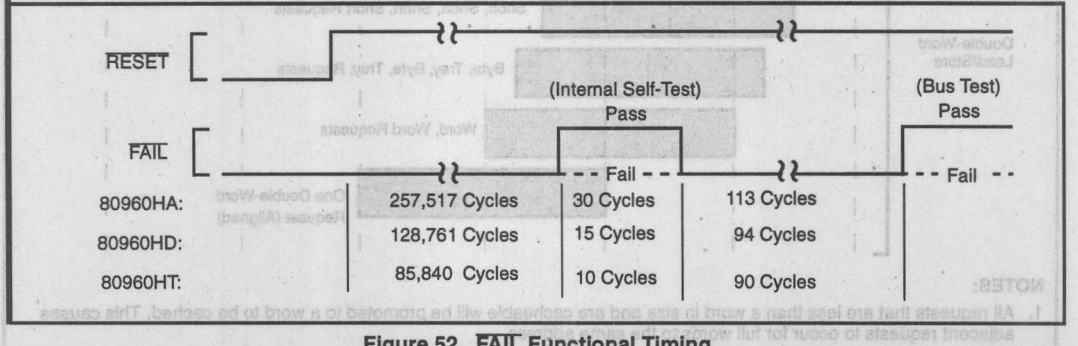
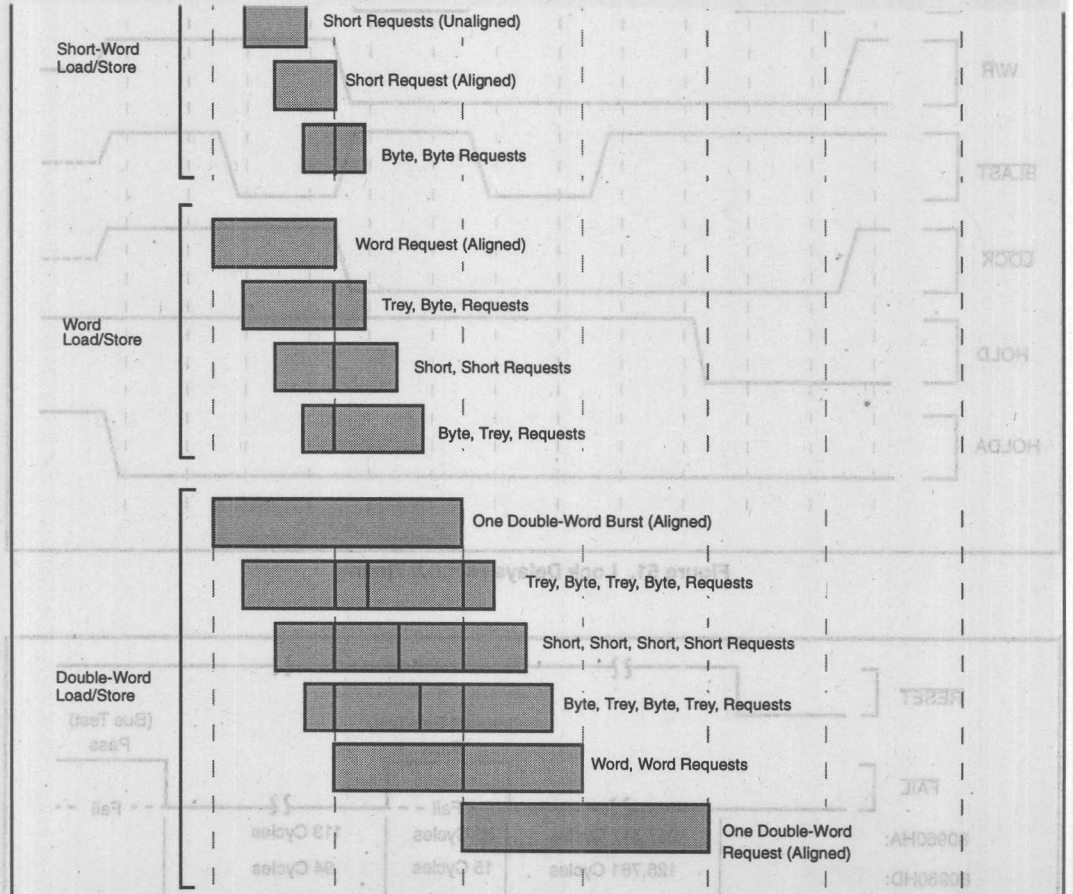


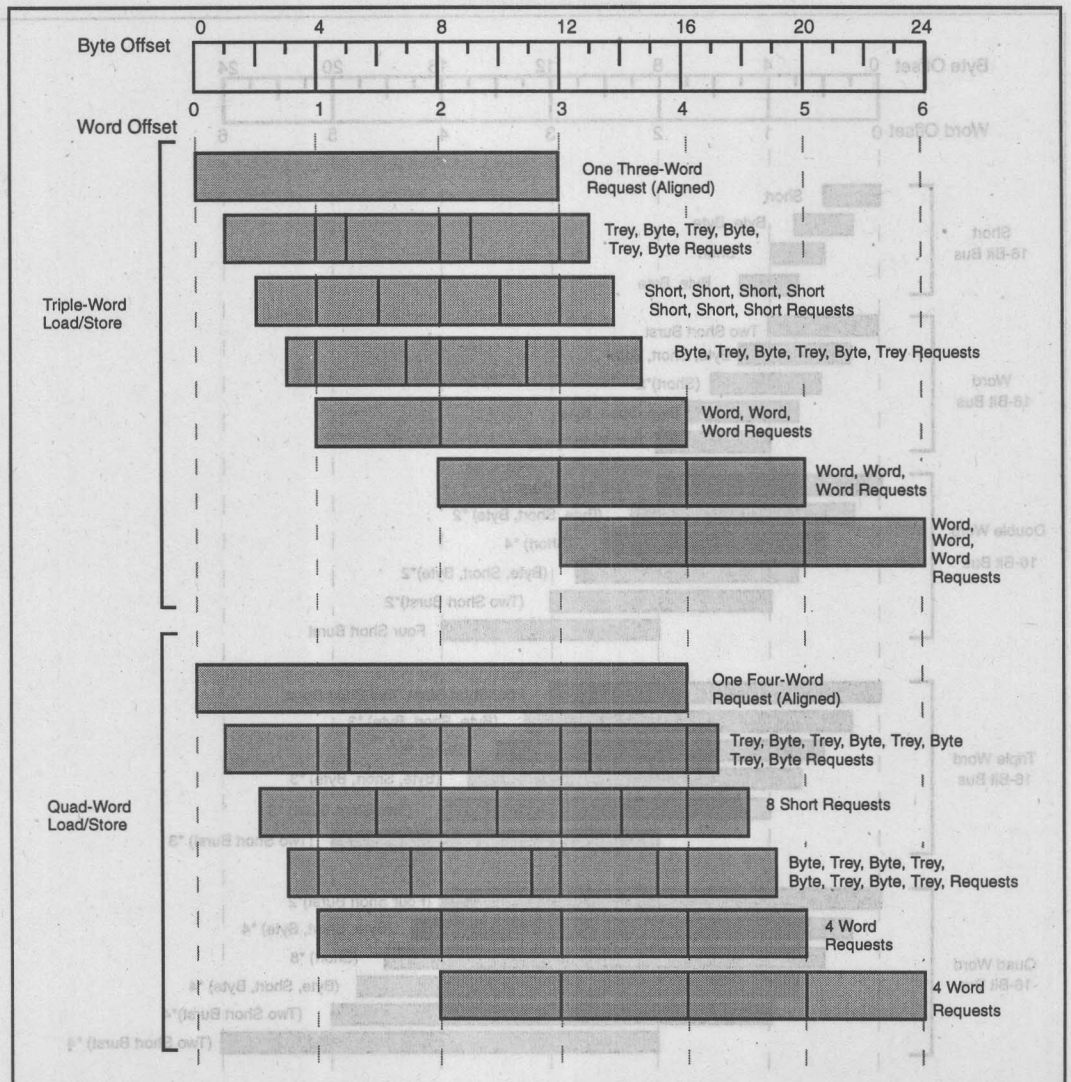
Figure 52. FAIL Functional Timing



NOTES:

1. All requests that are less than a word in size and are cacheable will be promoted to a word to be cached. This causes adjacent requests to occur for full words to the same address.

Figure 53. A Summary of Aligned and Unaligned Transfers for 32-Bit Regions



NOTES:

1. All requests that are less than a word in size and are cacheable will be promoted to a word to be cached. This causes adjacent requests to occur for full words to the same address.

Figure 54. A Summary of Aligned and Unaligned Transfers for 32-Bit Regions (Continued)

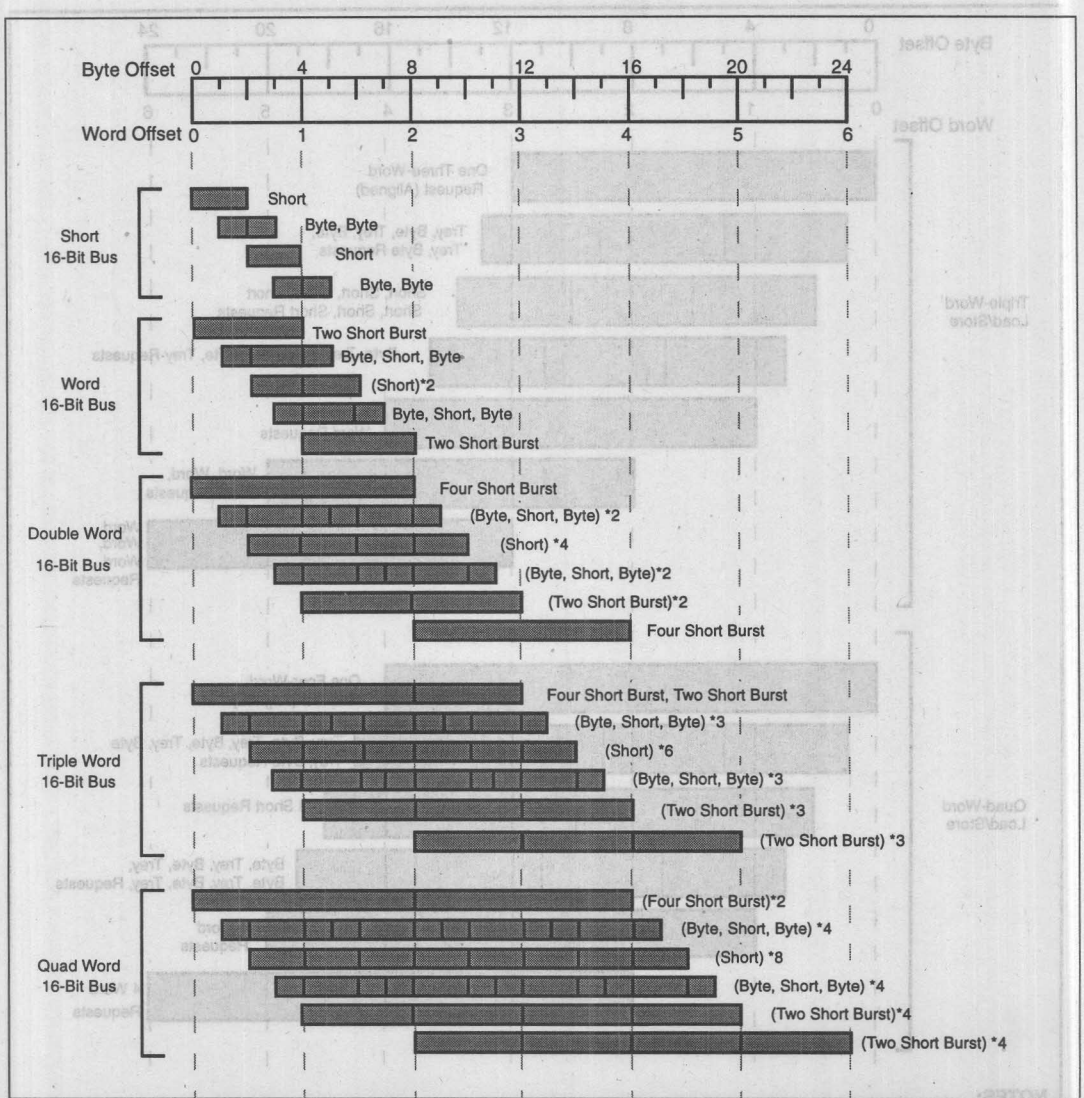


Figure 55. A Summary of Aligned and Unaligned Transfers for 16-Bit Bus

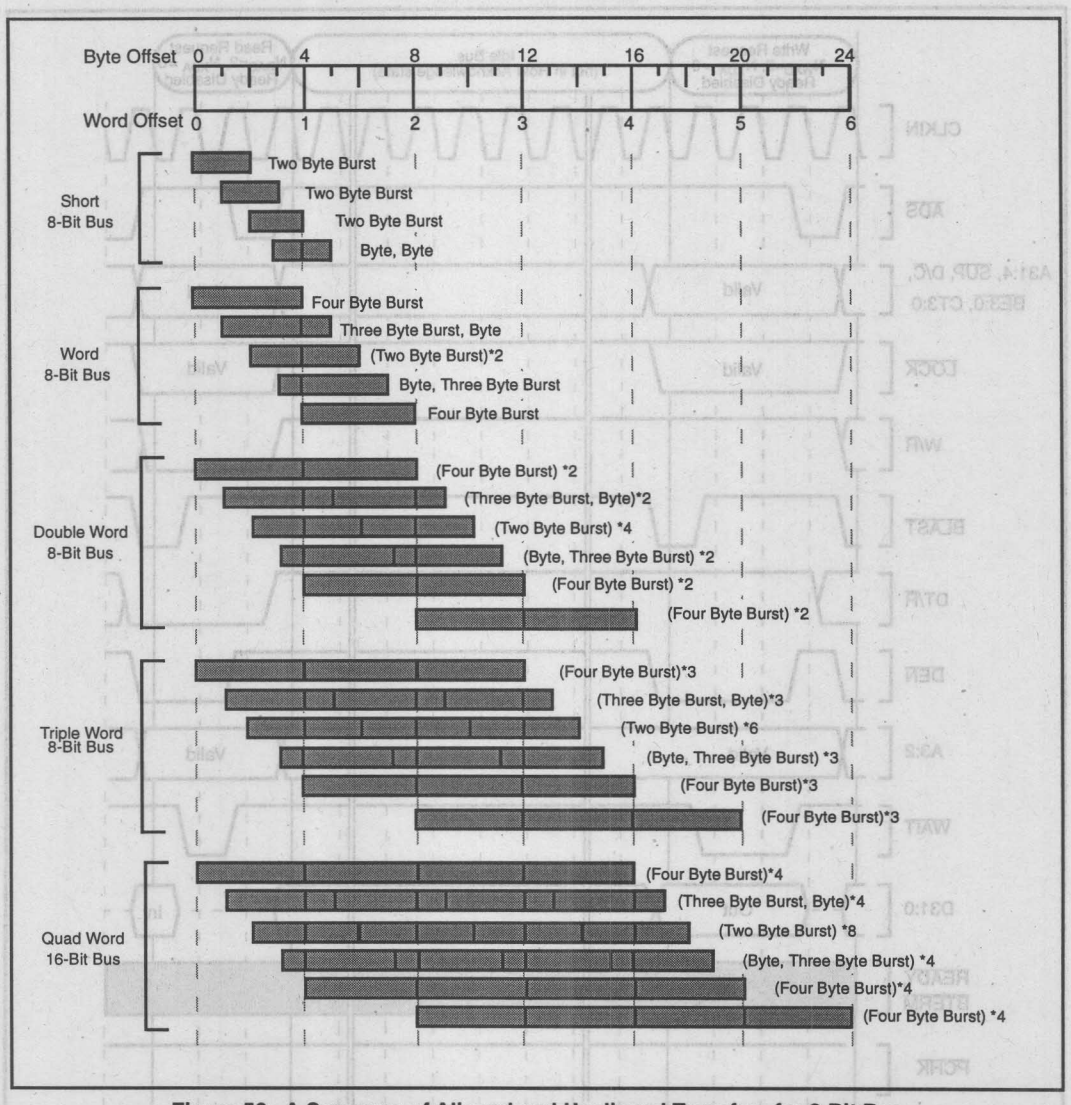


Figure 56. A Summary of Aligned and Unaligned Transfers for 8-Bit Bus

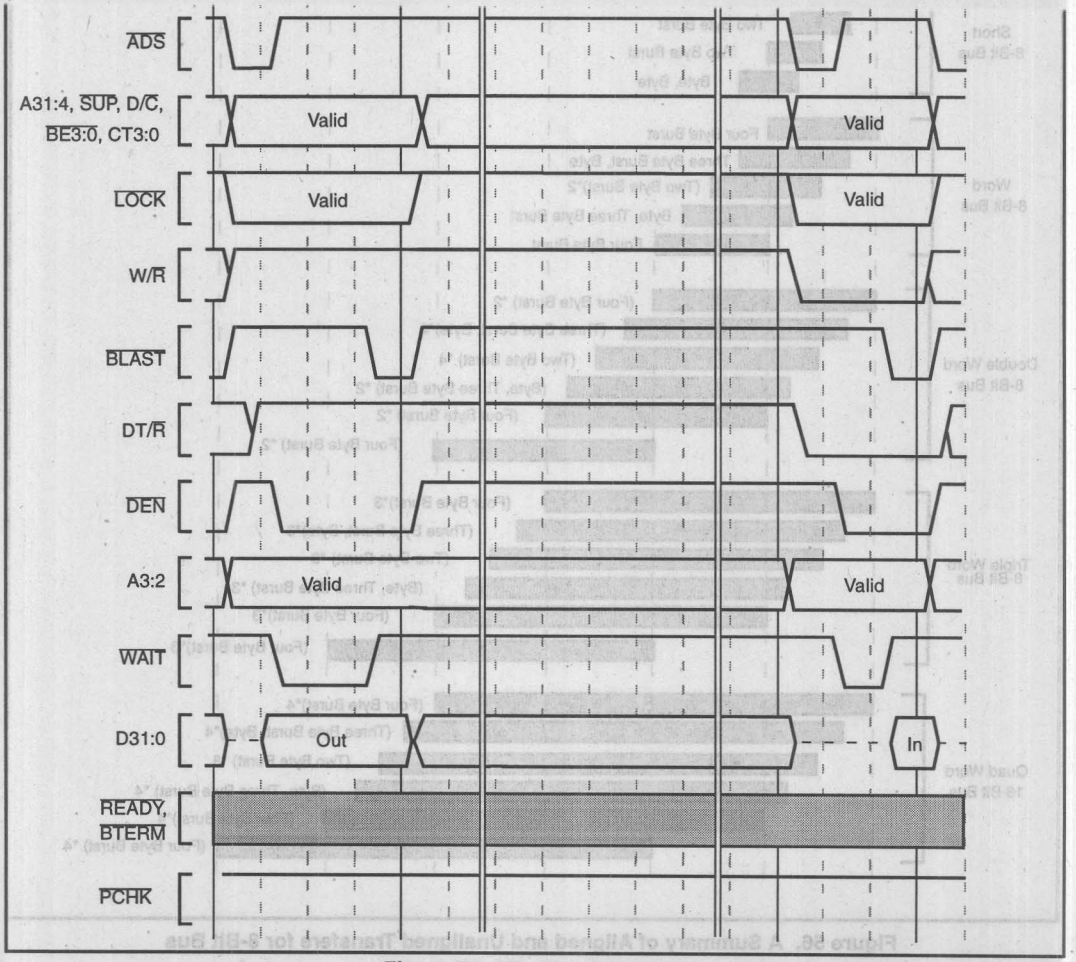
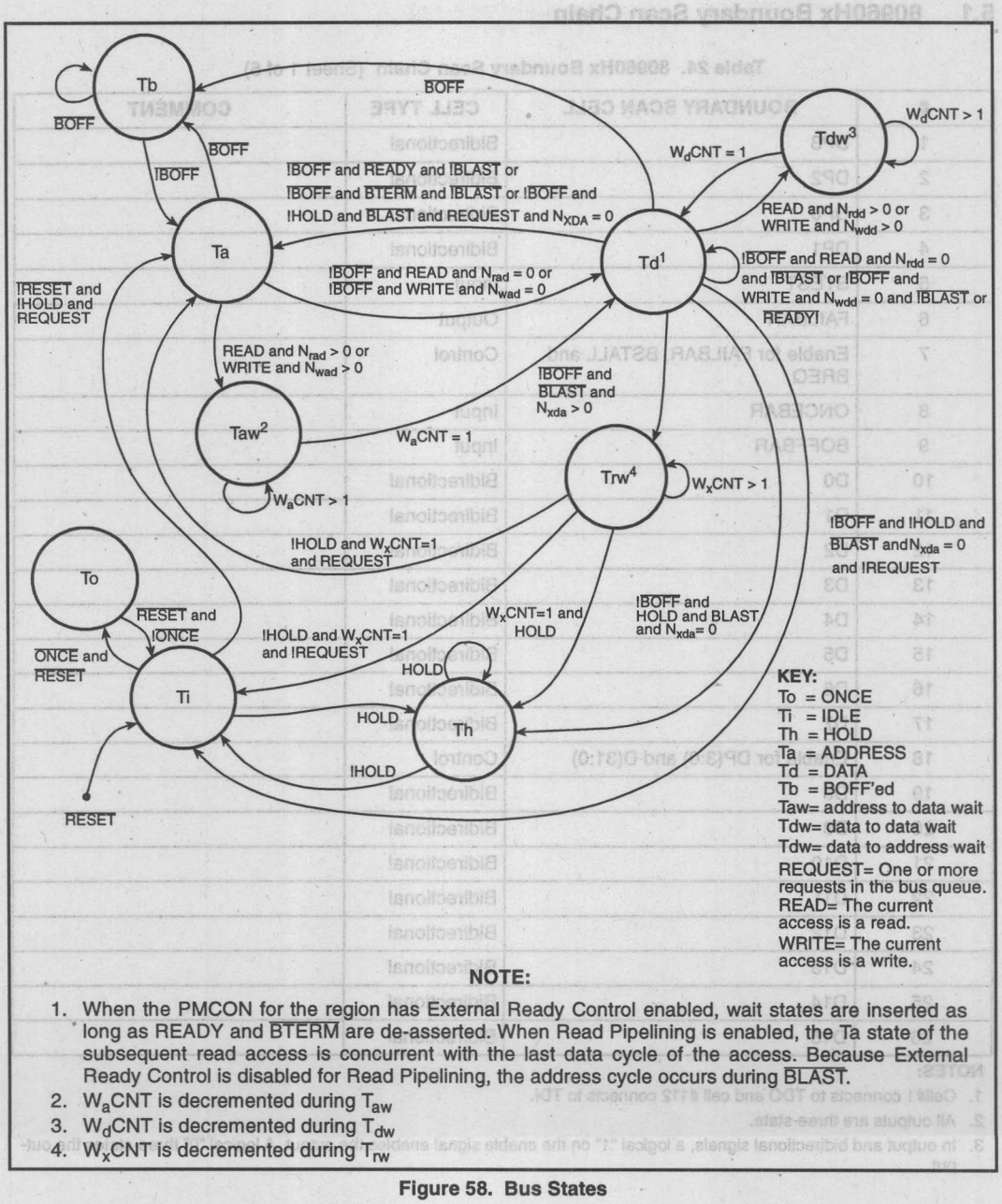


Figure 57. Idle Bus Operation



NOTE:

1. When the PMCON for the region has External Ready Control enabled, wait states are inserted as long as READY and BTERM are de-asserted. When Read Pipelining is enabled, the Ta state of the subsequent read access is concurrent with the last data cycle of the access. Because External Ready Control is disabled for Read Pipelining, the address cycle occurs during BLAST.
2. W_aCNT is decremented during T_{aw}
3. W_dCNT is decremented during T_{dw}
4. W_xCNT is decremented during T_{rw}

Figure 58. Bus States

5.1 80960Hx Boundary Scan Chain

Table 24. 80960Hx Boundary Scan Chain (Sheet 1 of 5)

#	BOUNDARY SCAN CELL	CELL TYPE	COMMENT
1	DP3	Bidirectional	
2	DP2	Bidirectional	
3	DP0	Bidirectional	
4	DP1	Bidirectional	
5	STEST	Input	
6	FAILBAR	Output	
7	Enable for FAILBAR, BSTALL and BREQ	Control	
8	ONCEBAR	Input	
9	BOFFBAR	Input	
10	D0	Bidirectional	
11	D1	Bidirectional	
12	D2	Bidirectional	
13	D3	Bidirectional	
14	D4	Bidirectional	
15	D5	Bidirectional	
16	D6	Bidirectional	
17	D7	Bidirectional	
18	Enable for DP(3:0) and D(31:0)	Control	
19	D8	Bidirectional	
20	D9	Bidirectional	
21	D10	Bidirectional	
22	D11	Bidirectional	
23	D12	Bidirectional	
24	D13	Bidirectional	
25	D14	Bidirectional	
26	D15	Bidirectional	

NOTES:
1. Cell#1 connects to TDO and cell #112 connects to TDI.
2. All outputs are three-state.
3. In output and bidirectional signals, a logical "1" on the enable signal enables the output. A logical "0" three-states the output.

Table 24. 80960Hx Boundary Scan Chain (Sheet 2 of 5)

#	BOUNDARY SCAN CELL	CELL TYPE	COMMENT
27	D16	Bidirectional	
28	D17	Bidirectional	
29	D18	Bidirectional	
30	D19	Bidirectional	
31	D20	Bidirectional	
32	D21	Bidirectional	
33	D22	Bidirectional	
34	D23	Bidirectional	
35	D24	Bidirectional	
36	D25	Bidirectional	
37	D26	Bidirectional	
38	D27	Bidirectional	
39	D28	Bidirectional	
40	D29	Bidirectional	
41	D30	Bidirectional	
42	D31	Bidirectional	
43	BTERMBAR	Input	
44	RDYBAR	Input	Appears as READYBAR in BSDL file.
45	HOLD	Input	
46	HOLDA	Output	
47	Enable for HOLDA control	Control	
48	ADSBAR	Output	
49	BE3BAR	Output	Appears as BEBAR(3:0) in BSDL file.
50	BE2BAR	Output	
51	BE1BAR	Output	
52	BE0BAR	Output	
53	BLASTBAR	Output	

NOTES:

1. Cell#1 connects to TDO and cell #112 connects to TDI.
2. All outputs are three-state.
3. In output and bidirectional signals, a logical "1" on the enable signal enables the output. A logical "0" three-states the output.

57	Enable for DTRBAR	Control	
58	WAITBAR	Output	
59	BSTALL	Output	
60	DATACODBAR	Output	Appears as DCBAR in BSDL file.
61	USERSUPBAR	Output	Appears as SUPBAR in BSDL file.
62	Enable for ADSBAR, BEBAR, BLASTBAR, DENBAR, WRRDBAR, WAITBAR, DCBAR, SUPBAR and LOCKBAR,	Control	
63	LOCKBAR	Output	
64	BREQ	Output	
65	A31	Output	
66	A30	Output	
67	A29	Output	
68	A28	Output	
69	A27	Output	
70	A26	Output	
71	A25	Output	
72	A24	Output	
73	A23	Output	
74	A22	Output	
75	A21	Output	
76	A20	Output	
77	A19	Output	
78	A18	Output	
79	A17	Output	
80	A16	Output	

NOTES:

1. Cell#1 connects to TDO and cell #112 connects to TDI.
2. All outputs are three-state.
3. In output and bidirectional signals, a logical "1" on the enable signal enables the output. A logical "0" three-states the output.

Table 24. 80960Hx Boundary Scan Chain (Sheet 4 of 5)

#	BOUNDARY SCAN CELL	CELL TYPE	COMMENT
81	Enable for A(31:0) and CT(3:0)	Control	
82	A15	Output	
83	A14	Output	
84	A13	Output	
85	A12	Output	
86	A11	Output	
87	A10	Output	
88	A9	Output	
89	A8	Output	
90	A7	Output	
91	A6	Output	
92	A5	Output	
93	A4	Output	
94	A3	Output	
95	A2	Output	
96	NMIBAR	Input	
97	XINT7BAR	Input	Appears as XINTBAR(7:0) in BSDL file.
98	XINT6BAR	Input	
99	XINT5BAR	Input	
100	XINT4BAR	Input	
101	XINT3BAR	Input	
102	XINT2BAR	Input	
103	XINT1BAR	Input	
104	XINT0BAR	Input	
105	RESETBAR	Input	
106	CLKIN	Input	
107	CT3	Output	Appears as CT(3:0) in BSDL file.

NOTES:

1. Cell#1 connects to TDO and cell #112 connects to TDI.
2. All outputs are three-state.
3. In output and bidirectional signals, a logical "1" on the enable signal enables the output. A logical "0" three-states the output.

Table 24. 80960Hx Boundary Scan Chain (Sheet 5 of 5)

#	BOUNDARY SCAN CELL	CELL TYPE	COMMENT
108	CT2	Output	
109	CT1	Output	
110	CT0	Output	
111	PCHK	Output	Appears as PCHKBAR in BSDL file.
112	PCHK enable	Control	

NOTES:
1. Cell#1 connects to TDO and cell #112 connects to TDI.
2. All outputs are three-state.
3. In output and bidirectional signals, a logical "1" on the enable signal enables the output. A logical "0" three-states the output.

5.2 Boundary Scan Description Language Example

Boundary-Scan Description Language (BSDL) example 14-2 meets the de facto standard means of describing essential features of ANSI/IEEE 1149.1-1993 compliant devices.

Example 5-1. Boundary-Scan Description Language (BSDL) for PGA Package Example (Sheet 1 of 9)

```
--Copyright Intel Corp. 1995
--*****************************************************************************
--Intel Corporation makes no warranty for the use of its products and assumes no
responsibility for any errors which may appear in this document nor does it make
a commitment to update the information contained herein.
--*****************************************************************************
--Boundary-Scan Description Language (BSDL Version 0.0) is a de-facto standard
means of describing essential features of ANSI/IEEE 1149.1-1990 compliant
devices. This language is under consideration by the IEEE for formal inclusion
within a supplement to the 1149.1-1990 standard. The generation of the
supplement entails an extensive IEEE review and a formal acceptance balloting
procedure which may change the resultant form of the language. Be aware that
this process may extend well into 1993, and at this time the IEEE does not
endorse or hold an opinion on the language.
*****************************************************************************
--
-- i960(R) Processor BSDL Model
```

Example 5-1. Boundary-Scan Description Language (BSDL) for PGA Package Example (Sheet 2 of 9)

```
-- Project code HA
-- File **NOT** verified electrically
-- ------------------------------------------------------------
-- Rev 0.7     18 Dec  1995  Updated for A-1 stepping.
-- Rev 0.6     08 Dec  1994
-- Rev 0.5     21 Nov  1994
-- Rev 0.4     31 Oct  1994
-- Rev 0.3     26 July 1994
-- Rev 0.2     22 June 1994
-- Rev 0.1     16 Mar  1994
-- Rev 0.0     30 Aug  1993
entity Ha_Processor is
   generic(PHYSICAL_PIN_MAP : string:= "PGA");

   port (A          : out     bit_vector(2 to 31);
         ADSBAR     : out     bit;
         BEBAR      : out     bit_vector(0 to 3);
         BLASTBAR   : out     bit;
         BOFFBAR    : in      bit;
         BREQ       : out     bit;
         BSTALL     : out     bit;
         BTERMBAR   : in      bit;
         CT         : out     bit_vector(0 to 3);
         CLKIN      : in      bit;
         D          : inout   bit_vector(0 to 31);
         DENBAR     : out     bit;
         DP         : inout   bit_vector(0 to 3);
         DTRBAR     : out     bit;
         DCBAR      : out     bit;
         FAILBAR    : out     bit;
         HOLD       : in      bit;
         HOLDA      : out     bit;
         LOCKBAR    : out     bit;
         NMIBAR     : in      bit;
         ONCEBAR    : in      bit;
         PCHKBAR    : out     bit;
         READYBAR   : in      bit;
         RESETBAR   : in      bit;
         STEST      : in      bit;
```

```
        TRST      : in      bit;
        WAITBAR   : out     bit;
        WRBAR     : out     bit;
        XINTBAR   : in      bit_vector(0 to 7);
        FIVEVREF  : linkage bit;
        VCCPLL    : linkage bit;
        VOLTDET   : out     bit;
        VCC1      : linkage bit_vector(0 to 23);
        VCC2      : linkage bit_vector(0 to 20);
        VSS1      : linkage bit_vector(0 to 25);
        VSS2      : linkage bit_vector(0 to 22);
        NC        : linkage bit_vector(0 to 4)

);

use STD_1149_1_1990.all;
use i960ha_a.all;


attribute PIN_MAP of Ha_Processor : entity is PHYSICAL_PIN_MAP;
 constant PGA:PIN_MAP_STRING :=
     "A         : (D16, D17, E16, E17, F17, G16, G17, H17, J17, K17,"&
     "               L17, L16, M17, N17, N16, P17, Q17, P16, P15, Q16,"&
     "               R17, R16, Q15, S17, R15, S16, Q14, R14, Q13, S15),"&
     "ADSBAR    : R06,"&
     "BEBAR     : (R09, S07, S06, S05),"&
     "BLASTBAR  : S08,"&
     "BOFFBAR   : B01,"&
     "BREQ      : R13,"&
     "BSTALL    : R12,"&
     "BTERMBAR  : R04,"&
     "CT        : (A11, A12, A13, A14),"&
     "CLKIN     : C13,"&
```

Example 5-1. Boundary-Scan Description Language (BSDL) for PGA Package Example (Sheet 4 of 9)

```
        "D          : (E03, C02, D02, C01, E02, D01, F02, E01, F01, G01,"&
        "                H02, H01, J01, K01, L02, L01, M01, N01, N02, P01,"&
        "                P02, Q01, P03, Q02, R01, S01, Q03, R02, Q04, S02,"&
        "                Q05, R03),"&
        "DENBAR     : S09,"&
        "DP         : (A03, B03, A04, B04),"&
        "DTRBAR     : S11,"&
        "DCBAR      : S13,"&
        "FAILBAR    : A02,"&
        "HOLD       : R05,"&
        "HOLDA      : S04,"&
        "LOCKBAR    : S14,"&
        "NMIBAR     : D15,"&
        "ONCEBAR    : C03,"&
        "PCHKBAR    : B08,"&
        "READYBAR   : S03,"&
        "RESETBAR   : A16,"&
        "STEST      : B02,"&
        "SUPBAR     : Q12,"&
        "TCK        : B05,"&
        "TDI        : A07,"&
        "TDO        : A08,"&
        "TMS        : B06,"&
        "TRST       : A06,"&
        "WAITBAR    : S12,"&
        "WRBAR      : S10,"&
        "XINTBAR    : (B15, A15, A17, B16, C15, B17, C16, C17),"&
        "FIVEVREF   : C05,"&
        "VOLTDET    : A05,"&
        "VCCPLL     : B10,"&
        "VCC1       : (M02, K02, J02, G02, N03, F03, C06, B07, B09, B11,"&
        "                B12, C14, E15, F16, H16, J16, K16, M16, N15, Q06,"&
        "                R07, R08, R10, R11),"&
        "VSS1       : (G03, H03, J03, K03, L03, M03, C07, C08, C09, C10,"&
        "                C11, C12, Q07, Q08, Q09, Q10, Q11, F15, G15, H15,"&
        "                J15, K15, L15, M15, A01, C04),"&
        "NC         : (A09, A10, B13, B14, D03)";
```

Example 5-1. Boundary-Scan Description Language (BSDL) for PGA Package Example (Sheet 5 of 9)

```
   attribute Tap_Scan_In     of TDI  : signal is true;
   attribute Tap_Scan_Mode   of TMS  : signal is true;
   attribute Tap_Scan_Out    of TDO  : signal is true;
   attribute Tap_Scan_Reset  of TRST : signal is true;
   attribute Tap_Scan_Clock  of TCK  : signal is (66.0e6, BOTH);

   attribute Instruction_Length of Ha_Processor: entity is 4;

   attribute Instruction_Opcode of Ha_Processor: entity is

      "BYPASS     (1111)," &
      "EXTEST     (0000)," &
      "SAMPLE     (0001)," &
      "IDCODE     (0010)," &
      "RUBIST     (0111)," &
      "CLAMP      (0100)," &
      "HIGHZ      (1000)," &
      "Reserved   (1011, 1100)";

   attribute Instruction_Capture of Ha_Processor: entity is "0001";

   attribute Instruction_Private of Ha_Processor: entity is "Reserved" ;

   attribute Idcode_Register of Ha_Processor: entity is
      "0001"                  &  --version,
      "1000100001000000"      &  --part number
      "00000001001"           &  --manufacturers identity
      "1";                       --required by the standard
   attribute Register_Access of Ha_Processor: entity is
     "Runbist[32]     (RUBIST)," &
     "Bypass          (CLAMP, HIGHZ)";
--{*************************************************************************}
--{  The first cell, cell 0, is closest to TDO                               }
--{  BC_1:Control, Output3  CBSC_1:Bidir  BC_4: Input, Clock                 }
--{*************************************************************************}
```

Example 5-1. Boundary-Scan Description Language (BSDL) for PGA Package Example (Sheet 6 of 9)

```
attribute Boundary_Cells of Ha_Processor: entity is "BC_4, BC_1, CBSC_1";
attribute Boundary_Length of Ha_Processor: entity is 112;
attribute Boundary_Register of Ha_Processor: entity is

   "0  (CBSC_1,  DP(3),     bidir,   X,  17, 1,  Z)," &
   "1  (CBSC_1,  DP(2),     bidir,   X,  17, 1,  Z)," &
   "2  (CBSC_1,  DP(0),     bidir,   X,  17, 1,  Z)," &
   "3  (CBSC_1,  DP(1),     bidir,   X,  17, 1,  Z)," &
   "4  (BC_4,    STEST,     input,   X)," &
   "5  (BC_1,    FAILBAR,   output3, X,   6,  1,  Z)," &
   "6  (BC_1,    *,         control, 1)," &
   "7  (BC_4,    ONCEBAR,   input,   X)," &
   "8  (BC_4,    BOFFBAR,   input,   X)," &
   "9  (CBSC_1,  D(0),      bidir,   X,  17,  1,  Z)," &
   "10 (CBSC_1,  D(1),      bidir,   X,  17,  1,  Z)," &
   "11 (CBSC_1,  D(2),      bidir,   X,  17,  1,  Z)," &
   "12 (CBSC_1,  D(3),      bidir,   X,  17,  1,  Z)," &
   "13 (CBSC_1,  D(4),      bidir,   X,  17,  1,  Z)," &
   "14 (CBSC_1,  D(5),      bidir,   X,  17,  1,  Z)," &
   "15 (CBSC_1,  D(6),      bidir,   X,  17,  1,  Z)," &
   "16 (CBSC_1,  D(7),      bidir,   X,  17,  1,  Z)," &
   "17 (BC_1,    *,         control, 1)," &
   "18 (CBSC_1,  D(8),      bidir,   X,  17,  1,  Z)," &
   "19 (CBSC_1,  D(9),      bidir,   X,  17,  1,  Z)," &
   "20 (CBSC_1,  D(10),     bidir,   X,  17,  1,  Z)," &
   "21 (CBSC_1,  D(11),     bidir,   X,  17,  1,  Z)," &
   "22 (CBSC_1,  D(12),     bidir,   X,  17,  1,  Z)," &
   "23 (CBSC_1,  D(13),     bidir,   X,  17,  1,  Z)," &
   "24 (CBSC_1,  D(14),     bidir,   X,  17,  1,  Z)," &
   "25 (CBSC_1,  D(15),     bidir,   X,  17,  1,  Z)," &
   "26 (CBSC_1,  D(16),     bidir,   X,  17,  1,  Z)," &
   "27 (CBSC_1,  D(17),     bidir,   X,  17,  1,  Z)," &
   "28 (CBSC_1,  D(18),     bidir,   X,  17,  1,  Z)," &
   "29 (CBSC_1,  D(19),     bidir,   X,  17,  1,  Z)," &
   "30 (CBSC_1,  D(20),     bidir,   X,  17,  1,  Z)," &
   "31 (CBSC_1,  D(21),     bidir,   X,  17,  1,  Z)," &
   "32 (CBSC_1,  D(22),     bidir,   X,  17,  1,  Z)," &
   "33 (CBSC_1,  D(23),     bidir,   X,  17,  1,  Z)," &
```

```
"40 (CBSC_1,   D(30),    bidir,   X, 17, 1,  Z)," &
"41 (CBSC_1,   D(31),    bidir,   X, 17, 1,  Z)," &
"42 (BC_4,     BTERMBAR,  input,   X)," &
"43 (BC_4,     READYBAR,  input,   X)," &
"44 (BC_4,     HOLD,      input,   X)," &
"45 (BC_1,     HOLDA,     output3, X, 46, 1,  Z)," &
"46 (BC_1,     *,         control, 1)," &
"47 (BC_1,     ADSBAR,    output3, X, 61, 1,  Z)," &
"48 (BC_1,     BEBAR(3),  output3, X, 61, 1,  Z)," &
"49 (BC_1,     BEBAR(2),  output3, X, 61, 1,  Z)," &
"50 (BC_1,     BEBAR(1),  output3, X, 61, 1,  Z)," &
"51 (BC_1,     BEBAR(0),  output3, X, 61, 1,  Z)," &
"52 (BC_1,     BLASTBAR,  output3, X, 61, 1,  Z)," &
"53 (BC_1,     DENBAR,    output3, X, 61, 1,  Z)," &
"54 (BC_1,     WRBAR,     output3, X, 61, 1,  Z)," &
"55 (BC_1,     DTRBAR,    output3, X, 56, 1,  Z)," &
"56 (BC_1,     *,         control, 1)," &
"57 (BC_1,     WAITBAR,   output3, X, 61, 1,  Z)," &
"58 (BC_1,     BSTALL,    output3, X, 6,  1,  Z)," &
"59 (BC_1,     DCBAR,     output3, X, 61, 1,  Z)," &
"60 (BC_1,     SUPBAR,    output3, X, 61, 1,  Z)," &
"61 (BC_1,     *,         control, 1)," &
"62 (BC_1,     LOCKBAR,   output3, X, 61, 1,  Z)," &
"63 (BC_1,     BREQ,      output3, X, 6,  1,  Z)," &
"64 (BC_1,     A(31),     output3, X, 80, 1,  Z)," &
"65 (BC_1,     A(30),     output3, X, 80, 1,  Z)," &
"66 (BC_1,     A(29),     output3, X, 80, 1,  Z)," &
"67 (BC_1,     A(28),     output3, X, 80, 1,  Z)," &
"68 (BC_1,     A(27),     output3, X, 80, 1,  Z)," &
"69 (BC_1,     A(26),     output3, X, 80, 1,  Z)," &
"70 (BC_1,     A(25),     output3, X, 80, 1,  Z)," &
"71 (BC_1,     A(24),     output3, X, 80, 1,  Z)," &
```

Example 5-1. Boundary-Scan Description Language (BSDL) for PGA Package Example (Sheet 8 of 9)

```
"72 (BC_1,  A(23),  output3,  X,  80, 1,  Z)," &
"73 (BC_1,  A(22),  output3,  X,  80, 1,  Z)," &
"74 (BC_1,  A(21),  output3,  X,  80, 1,  Z)," &
"75 (BC_1,  A(20),  output3,  X,  80, 1,  Z)," &
"76 (BC_1,  A(19),  output3,  X,  80, 1,  Z)," &
"77 (BC_1,  A(18),  output3,  X,  80, 1,  Z)," &
"78 (BC_1,  A(17),  output3,  X,  80, 1,  Z)," &
"79 (BC_1,  A(16),  output3,  X,  80, 1,  Z)," &
"80 (BC_1,  *,      control,  1)," &
"81 (BC_1,  A(15),  output3,  X,  80, 1,  Z)," &
"82 (BC_1,  A(14),  output3,  X,  80, 1,  Z)," &
"83 (BC_1,  A(13),  output3,  X,  80, 1,  Z)," &
"84 (BC_1,  A(12),  output3,  X,  80, 1,  Z)," &
"85 (BC_1,  A(11),  output3,  X,  80, 1,  Z)," &
"86 (BC_1,  A(10),  output3,  X,  80, 1,  Z)," &
"87 (BC_1,  A(9),   output3,  X,  80, 1,  Z)," &
"88 (BC_1,  A(8),   output3,  X,  80, 1,  Z)," &
"89 (BC_1,  A(7),   output3,  X,  80, 1,  Z)," &
"90 (BC_1,  A(6),   output3,  X,  80, 1,  Z)," &
"91 (BC_1,  A(5),   output3,  X,  80, 1,  Z)," &
"92 (BC_1,  A(4),   output3,  X,  80, 1,  Z)," &
"93 (BC_1,  A(3),   output3,  X,  80, 1,  Z)," &
"94 (BC_1,  A(2),   output3,  X,  80, 1,  Z)," &
"95 (BC_4,  NMIBAR,       input,  X)," &
"96 (BC_4,  XINTBAR(7),   input,  X)," &
"97 (BC_4,  XINTBAR(6),   input,  X)," &
"98 (BC_4,  XINTBAR(5),   input,  X)," &
"99 (BC_4,  XINTBAR(4),   input,  X)," &
"100(BC_4,  XINTBAR(3),   input,  X)," &
"101(BC_4,  XINTBAR(2),   input,  X)," &
"102(BC_4,  XINTBAR(1),   input,  X)," &
"103(BC_4,  XINTBAR(0),   input,  X)," &
"104(BC_4,  RESETBAR,     input,  X)," &
"105(BC_4,  CLKIN,        input,  X)," &
"106(BC_1,  CT(3),        output3,  X,  80, 1,  Z)," &
```

Example 5-1. Boundary-Scan Description Language (BSDL) for PGA Package Example (Sheet 9 of 9)

```
      "107(BC_1,  CT(2),      output3,  X,   80, 1, Z)," &
      "108(BC_1,  CT(1),      output3,  X,   80, 1, Z)," &
      "109(BC_1,  CT(0),      output3,  X,   80, 1, Z)," &
      "110(BC_1,  PCHKBAR,    output3,  X,  111, 1, Z)," &
      "111(BC_1,  *,          control,  1)";
end Ha_Processor;
```

Example 5-2. Boundary-Scan Description Language (BSDL) for PQ2 Package Example (Sheet 1 of 8)

```
-- Copyright Intel Corporation 1995, 1996
--*******************************************************************************
--Intel Corporation makes no warranty for the use of its products and assumes no
responsibility for any errors which may appear in this document nor does it make
a commitment to update the information contained herein.
--*******************************************************************************
--Boundary-Scan Description Language (BSDL Version 0.0) is a de-facto
--standard means of describing essential features of ANSI/IEEE 1149.1-1990
compliant devices.  This language is under consideration by the IEEE for formal
inclusion within a supplement to the 1149.1-1990 standard.  The generation of
the supplement entails an extensive IEEE review and a formal acceptance
balloting procedure which may change the resultant form of the language.  Be
aware that this process may extend well into 1993, and at this time the IEEE
does not endorse or hold an opinion on the language.
-- i960(R) Processor BSDL Model
-- Project code HA
-- File **NOT** verified electrically
--  ------------------------------------------------------------
-- Rev 0.8      4 Apr  1996  Changed for PQ2 Package
-- Rev 0.7     18 Dec  1995  Updated for A-1 stepping.
-- Rev 0.6     08 Dec  1994
-- Rev 0.5     21 Nov  1994
-- Rev 0.4     31 Oct  1994
-- Rev 0.3     26 July 1994
-- Rev 0.2     22 June 1994
-- Rev 0.1     16 Mar  1994
-- Rev 0.0     30 Aug  1993
```

Example 5-2. Boundary-Scan Description Language (BSDL) for PQ2 Package Example (Sheet 2 of 8)

```
entity Ha_Processor is
   generic(PHYSICAL_PIN_MAP : string:= "PQ2");
   port (A          : out     bit_vector(2 to 31);
         ADSBAR     : out     bit;
         BEBAR      : out     bit_vector(0 to 3);
         BLASTBAR   : out     bit;
         BOFFBAR    : in      bit;
         BREQ       : out     bit;
         BSTALL     : out     bit;
         BTERMBAR   : in      bit;
         CT         : out     bit_vector(0 to 3);
         CLKIN      : in      bit;
         D          : inout   bit_vector(0 to 31);
         DENBAR     : out     bit;
         DP         : inout   bit_vector(0 to 3);
         DTRBAR     : out     bit;
         DCBAR      : out     bit;
         FAILBAR    : out     bit;
         HOLD       : in      bit;
         HOLDA      : out     bit;
         LOCKBAR    : out     bit;
         NMIBAR     : in      bit;
         ONCEBAR    : in      bit;
         PCHKBAR    : out     bit;
         READYBAR   : in      bit;
         RESETBAR   : in      bit;
         STEST      : in      bit;
         SUPBAR     : out     bit;
         TCK        : in      bit;
         TDI        : in      bit;
         TDO        : out     bit;
         TMS        : in      bit;
         TRST       : in      bit;
         WAITBAR    : out     bit;
         WRBAR      : out     bit;
         XINTBAR    : in      bit_vector(0 to 7);
         FIVEVREF   : linkage bit;
         VCCPLL     : linkage bit;
```

Example 5-2. Boundary-Scan Description Language (BSDL) for PQ2 Package Example (Sheet 3 of 8)

```
        VCC1        : linkage bit_vector(0 to 23);
        VCC2        : linkage bit_vector(0 to 23);
        VSS1        : linkage bit_vector(0 to 23);
        VSS2        : linkage bit_vector(0 to 23)

);

use STD_1149_1_1990.all;
use i960ha_a.all;


attribute PIN_MAP of Ha_Processor : entity is PHYSICAL_PIN_MAP;
 constant PQ2:PIN_MAP_STRING :=

      "A          : (151, 150, 147, 146, 145, 144, 141, 140, 139, 138,"&
      "                135, 134, 133, 132, 127, 126, 125, 124, 121, 120,"&
      "                119, 118, 113, 112, 111, 110, 107, 106, 105, 104),"&
      "ADSBAR     : 77,"&
      "BEBAR      : (83, 82, 79, 78),"&
      "BLASTBAR   : 84,"&
      "BOFFBAR    : 10,"&
      "BREQ       : 100,"&
      "BSTALL     : 91,"&
      "BTERMBAR   : 67,"&
      "CT         : (183, 182, 181, 180),"&
      "CLKIN      : 175,"&
      "D          : (12, 13, 14, 15, 20, 21, 22, 23, 26, 27, 28, 29,"&
      "               34, 35, 36, 37, 39, 40, 41, 42, 45, 50, 51, 52,"&
      "               54, 55, 56, 57, 61, 62, 63, 64),"&
      "DENBAR     : 85,"&
      "DP         : (206, 207, 203, 202),"&
      "DTRBAR     : 89,"&
      "DCBAR      : 96,"&
      "FAILBAR    : 5,"&
      "HOLD       : 69,"&
      "HOLDA      : 72,"&
      "LOCKBAR    : 99,"&
      "NMIBAR     : 159,"&
```

Example 5-2. Boundary-Scan Description Language (BSDL) for PQ2 Package Example (Sheet 4 of 8)

```
        "ONCEBAR    : 6,"&
        "PCHKBAR    : 189,"&
        "READYBAR   : 68,"&
        "RESETBAR   : 174,"&
        "STEST      : 208,"&
        "SUPBAR     : 97,"&
        "TCK        : 194,"&
        "TDI        : 191,"&
        "TDO        : 188,"&
        "TMS        : 192,"&
        "TRST       : 193,"&
        "WAITBAR    : 90,"&
        "WRBAR      : 88,"&
        "XINTBAR    : (169, 168, 167, 166, 163, 162, 161, 160),"&
        "FIVEVREF   : 197,"&
        "VCCPLL     : 177,"&
        "VCC1       : (1, 4, 9, 11, 17, 19, 25, 31, 33, 38, 44, 46,"&
        "             49, 59, 60, 66, 71, 74, 76, 81, 87, 92, 95, 101),"&
        "VCC2       : (102, 109, 115, 117, 123, 128, 131, 137, 143, 149,"&
        "             153, 154, 158, 165, 171, 173, 176, 179, 185, 187,"&
        "             196, 199, 201, 204),"&
        "VSS1       : (2, 3, 7, 8, 16, 18, 24, 30, 32, 43, 47, 48,"&
        "             53, 58, 65, 70, 73, 75, 80, 86, 93, 94, 98, 103),"&
        "VSS2       : (108, 114, 116, 122, 129, 130, 136, 142, 148, 152,"&
        "             155, 156, 157, 164, 170, 172, 178, 184, 186, 190,"&
        "             195, 198, 200, 205)";

  attribute Tap_Scan_In     of  TDI   : signal is true;
  attribute Tap_Scan_Mode   of  TMS   : signal is true;
  attribute Tap_Scan_Out    of  TDO   : signal is true;
  attribute Tap_Scan_Reset  of  TRST  : signal is true;
  attribute Tap_Scan_Clock  of  TCK   : signal is (66.0e6, BOTH);

  attribute Instruction_Length of Ha_Processor: entity is 4;

  attribute Instruction_Opcode of Ha_Processor: entity is
```

```
      "Reserved   (1011, 1100)";

   attribute Instruction_Capture of Ha_Processor: entity is "0001";

   attribute Instruction_Private of Ha_Processor: entity is "Reserved" ;

   attribute Idcode_Register of Ha_Processor: entity is
      "0001"                  &  version,
      "1000100001000000"      &  part number
      "00000001001"           &  manufacturers identity
      "1";                       required by the standard

   attribute Register_Access of Ha_Processor: entity is
     "Runbist[32]     (RUBIST)," &
     "Bypass          (CLAMP, HIGHZ)";
*****************************************************************************
{  The first cell, cell 0, is closest to TDO                                  }
{  BC_1:Control, Output3  CBSC_1:Bidir  BC_4: Input, Clock                    }
*****************************************************************************

   attribute Boundary_Cells of Ha_Processor: entity is "BC_4, BC_1, CBSC_1";
   attribute Boundary_Length of Ha_Processor: entity is 112;
   attribute Boundary_Register of Ha_Processor: entity is

      "0   (CBSC_1,  DP(3),      bidir,   X,  17, 1,  Z)," &
      "1   (CBSC_1,  DP(2),      bidir,   X,  17, 1,  Z)," &
      "2   (CBSC_1,  DP(0),      bidir,   X,  17, 1,  Z)," &
      "3   (CBSC_1,  DP(1),      bidir,   X,  17, 1,  Z)," &
      "4   (BC_4,  STEST,        input,   X)," &
      "5   (BC_1,  FAILBAR,      output3, X,  6,  1,  Z)," &
```

Example 5-2. Boundary-Scan Description Language (BSDL) for PQ2 Package Example (Sheet 6 of 8)

```
"6  (BC_1,    *,          control, 1),"  &
"7  (BC_4,    ONCEBAR,    input,   X),"  &
"8  (BC_4,    BOFFBAR,    input,   X),"  &
"9  (CBSC_1,  D(0),       bidir,   X,  17,  1,  Z),"  &
"10 (CBSC_1,  D(1),       bidir,   X,  17,  1,  Z),"  &
"11 (CBSC_1,  D(2),       bidir,   X,  17,  1,  Z),"  &
"12 (CBSC_1,  D(3),       bidir,   X,  17,  1,  Z),"  &
"13 (CBSC_1,  D(4),       bidir,   X,  17,  1,  Z),"  &
"14 (CBSC_1,  D(5),       bidir,   X,  17,  1,  Z),"  &
"15 (CBSC_1,  D(6),       bidir,   X,  17,  1,  Z),"  &
"16 (CBSC_1,  D(7),       bidir,   X,  17,  1,  Z),"  &
"17 (BC_1,    *,          control, 1),"  &
"18 (CBSC_1,  D(8),       bidir,   X,  17,  1,  Z),"  &
"19 (CBSC_1,  D(9),       bidir,   X,  17,  1,  Z),"  &
"20 (CBSC_1,  D(10),      bidir,   X,  17,  1,  Z),"  &
"21 (CBSC_1,  D(11),      bidir,   X,  17,  1,  Z),"  &
"22 (CBSC_1,  D(12),      bidir,   X,  17,  1,  Z),"  &
"23 (CBSC_1,  D(13),      bidir,   X,  17,  1,  Z),"  &
"24 (CBSC_1,  D(14),      bidir,   X,  17,  1,  Z),"  &
"25 (CBSC_1,  D(15),      bidir,   X,  17,  1,  Z),"  &
"26 (CBSC_1,  D(16),      bidir,   X,  17,  1,  Z),"  &
"27 (CBSC_1,  D(17),      bidir,   X,  17,  1,  Z),"  &
"28 (CBSC_1,  D(18),      bidir,   X,  17,  1,  Z),"  &
"29 (CBSC_1,  D(19),      bidir,   X,  17,  1,  Z),"  &
"30 (CBSC_1,  D(20),      bidir,   X,  17,  1,  Z),"  &
"31 (CBSC_1,  D(21),      bidir,   X,  17,  1,  Z),"  &
"32 (CBSC_1,  D(22),      bidir,   X,  17,  1,  Z),"  &
"33 (CBSC_1,  D(23),      bidir,   X,  17,  1,  Z),"  &
"34 (CBSC_1,  D(24),      bidir,   X,  17,  1,  Z),"  &
"35 (CBSC_1,  D(25),      bidir,   X,  17,  1,  Z),"  &
"36 (CBSC_1,  D(26),      bidir,   X,  17,  1,  Z),"  &
"37 (CBSC_1,  D(27),      bidir,   X,  17,  1,  Z),"  &
"38 (CBSC_1,  D(28),      bidir,   X,  17,  1,  Z),"  &
"39 (CBSC_1,  D(29),      bidir,   X,  17,  1,  Z),"  &
"40 (CBSC_1,  D(30),      bidir,   X,  17,  1,  Z),"  &
```

Example 5-2. Boundary-Scan Description Language (BSDL) for PQ2 Package Example (Sheet 7 of 8)

```
"41 (CBSC_1,  D(31),    bidir,    X,  17, 1,  Z)," &
"42 (BC_4,  BTERMBAR,   input,    X)," &
"43 (BC_4,  READYBAR,   input,    X)," &
"44 (BC_4,  HOLD,       input,    X)," &
"45 (BC_1,  HOLDA,      output3,  X,  46, 1,  Z)," &
"46 (BC_1,  *,          control,  1)," &
"47 (BC_1,  ADSBAR,     output3,  X,  61, 1,  Z)," &
"48 (BC_1,  BEBAR(3),   output3,  X,  61, 1,  Z)," &
"49 (BC_1,  BEBAR(2),   output3,  X,  61, 1,  Z)," &
"50 (BC_1,  BEBAR(1),   output3,  X,  61, 1,  Z)," &
"51 (BC_1,  BEBAR(0),   output3,  X,  61, 1,  Z)," &
"52 (BC_1,  BLASTBAR,   output3,  X,  61, 1,  Z)," &
"53 (BC_1,  DENBAR,     output3,  X,  61, 1,  Z)," &
"54 (BC_1,  WRBAR,      output3,  X,  61, 1,  Z)," &
"55 (BC_1,  DTRBAR,     output3,  X,  56, 1,  Z)," &
"56 (BC_1,  *,          control,  1)," &
"57 (BC_1,  WAITBAR,    output3,  X,  61, 1,  Z)," &
"58 (BC_1,  BSTALL,     output3,  X,  6,  1,  Z)," &
"59 (BC_1,  DCBAR,      output3,  X,  61, 1,  Z)," &
"60 (BC_1,  SUPBAR,     output3,  X,  61, 1,  Z)," &
"61 (BC_1,  *,          control,  1)," &
"62 (BC_1,  LOCKBAR,    output3,  X,  61, 1,  Z)," &
"63 (BC_1,  BREQ,       output3,  X,  6,  1,  Z)," &
"64 (BC_1,  A(31),      output3,  X,  80, 1,  Z)," &
"65 (BC_1,  A(30),      output3,  X,  80, 1,  Z)," &
"66 (BC_1,  A(29),      output3,  X,  80, 1,  Z)," &
"67 (BC_1,  A(28),      output3,  X,  80, 1,  Z)," &
"68 (BC_1,  A(27),      output3,  X,  80, 1,  Z)," &
"69 (BC_1,  A(26),      output3,  X,  80, 1,  Z)," &
"70 (BC_1,  A(25),      output3,  X,  80, 1,  Z)," &
"71 (BC_1,  A(24),      output3,  X,  80, 1,  Z)," &
"72 (BC_1,  A(23),      output3,  X,  80, 1,  Z)," &
"73 (BC_1,  A(22),      output3,  X,  80, 1,  Z)," &
"74 (BC_1,  A(21),      output3,  X,  80, 1,  Z)," &
"75 (BC_1,  A(20),      output3,  X,  80, 1,  Z)," &
```

Example 5-2. Boundary-Scan Description Language (BSDL) for PQ2 Package Example (Sheet 8 of 8)

```
      "76 (BC_1,  A(19),      output3,  X,  80, 1,  Z)," &
      "77 (BC_1,  A(18),      output3,  X,  80, 1,  Z)," &
      "78 (BC_1,  A(17),      output3,  X,  80, 1,  Z)," &
      "79 (BC_1,  A(16),      output3,  X,  80, 1,  Z)," &
      "80 (BC_1,  *,          control,  1)," &
      "81 (BC_1,  A(15),      output3,  X,  80, 1,  Z)," &
      "82 (BC_1,  A(14),      output3,  X,  80, 1,  Z)," &
      "83 (BC_1,  A(13),      output3,  X,  80, 1,  Z)," &
      "84 (BC_1,  A(12),      output3,  X,  80, 1,  Z)," &
      "85 (BC_1,  A(11),      output3,  X,  80, 1,  Z)," &
      "86 (BC_1,  A(10),      output3,  X,  80, 1,  Z)," &
      "87 (BC_1,  A(9),       output3,  X,  80, 1,  Z)," &
      "88 (BC_1,  A(8),       output3,  X,  80, 1,  Z)," &
      "89 (BC_1,  A(7),       output3,  X,  80, 1,  Z)," &
      "90 (BC_1,  A(6),       output3,  X,  80, 1,  Z)," &
      "91 (BC_1,  A(5),       output3,  X,  80, 1,  Z)," &
      "92 (BC_1,  A(4),       output3,  X,  80, 1,  Z)," &
      "93 (BC_1,  A(3),       output3,  X,  80, 1,  Z)," &
      "94 (BC_1,  A(2),       output3,  X,  80, 1,  Z)," &
      "95 (BC_4,  NMIBAR,     input,    X)," &
      "96 (BC_4,  XINTBAR(7), input,    X)," &
      "97 (BC_4,  XINTBAR(6), input,    X)," &
      "98 (BC_4,  XINTBAR(5), input,    X)," &
      "99 (BC_4,  XINTBAR(4), input,    X)," &
      "100(BC_4,  XINTBAR(3), input,    X)," &
      "101(BC_4,  XINTBAR(2), input,    X)," &
      "102(BC_4,  XINTBAR(1), input,    X)," &
      "103(BC_4,  XINTBAR(0), input,    X)," &
      "104(BC_4,  RESETBAR,   input,    X)," &
      "105(BC_4,  CLKIN,      input,    X)," &
      "106(BC_1,  CT(3),      output3,  X,  80, 1,  Z)," &
      "107(BC_1,  CT(2),      output3,  X,  80, 1,  Z)," &
      "108(BC_1,  CT(1),      output3,  X,  80, 1,  Z)," &
      "109(BC_1,  CT(0),      output3,  X,  80, 1,  Z)," &
      "110(BC_1,  PCHKBAR,    output3,  X, 111,1,  Z)," &
      "111(BC_1,  *,          control,  1)";
end Ha_Processor;
```

	HOLDA pin definition during BOFF mode corrected. Specified that $\overline{TRST}$ can be tied low when not used. Also changed the $\overline{TRST}$ pull-up resistor to a 65 kΩ resistor. Clarified that the $\overline{DEN}$ remains asserted for an entire bus request, even when that request spans several bus accesses.
Table 11, Maximum TA at Various Airflows in °C (PGA Package Only) (pg. 23)	Revised all table values.
Table 13, Maximum TA at Various Airflows in °C (PQ2 Package Only) (pg. 24)	Revised all table values and added data for 16 MHz clock within the Core 2X Bus Clock data.
Table 14, 80960Hx 208-Pin PQ2 Package Thermal Characteristics (pg. 24)	Added thermal data for PQ2 package. Changed title to refer to 208-pin package.
Table 17, Device ID Version Numbers for Different Steppings (pg. 26)	New table.
Table 18, Operating Conditions (pg. 27)	Changed Min and Max data for V_{CC} and VCC5.
Table 20, 80960Hx DC Characteristics (pg. 29)	Added 3.3 V and 5 V AC and DC specifications based on design simulations. Revised values for I_{LI}, I_{LO}, I_{CC} Power Supply, I_{CC} Thermal, I_{CC} Test, and R_{PU}. Increased I_{OL} for $\overline{FAIL}$ pin. Corrected sign for I_{LI} and increased value for pins with resistor pull-ups. Added categories under I_{CC}. Removed I_{HALT}. Added ONCE parameter to R_{PU} and reduced resistance values. Added C_{IN} for PQ2 package.
Table 21, 80960Hx AC Characteristics (pg. 31)	Changed max value for T_F. Changed min value for T. Redefined T_{CS}. Revised synchronous and asynchronous timings based on silicon characterizations.
Table 23, 80960Hx Boundary Scan Test Signal Timings (pg. 33)	Added values for T_{BSF} min and T_{BSC} max.

Table 25. Data Sheet Version -003 to -004 Revision History (Sheet 2 of 2)

Section	Description
Section 4.6.1, AC Test Conditions (pg. 34)	Corrected reference to Figure 25.
Figure 22, Rise and Fall Time Derating at 85°C and Minimum VCC (pg. 41) Figure 23, ICC Active (Power Supply) vs. Frequency (pg. 41) Figure 24, ICC Active (Thermal) vs. Frequency (pg. 42) Figure 26, Output Delay vs. Temperature (pg. 43) Figure 27, Output Hold Times vs. Temperature (pg. 43) Figure 28, Output Delay vs. VCC (pg. 44)	New figures.
Figure 29, Cold Reset Waveform (pg. 45) Figure 30, Warm Reset Waveform (pg. 46)	Added values for $\overline{\text{RESET}}$ periods.
Figure 31, Entering ONCE Mode (pg. 47)	Added values for V_{CC} periods and indicated that CLKIN must be active.
Figures 33 through 39 (pages 49 through 55)	Revised $\overline{\text{DEN}}$ waveform.
Figures 38, 39, and 40 (pages 54 through 56)	Revised $\overline{\text{WAIT}}$ waveforms.
Figure 40, Burst, Non-Pipelined Read Request with Wait States, 8-Bit Bus (pg. 56)	Changed Burst and Non-Burst identifiers.
Figure 49, BOFF Functional Timing (pg. 65)	Moved Begin Request and End Request terminals outward.
Figure 58, Bus States (pg. 73)	Added $\overline{\text{READY}}$ signal to ADDRESS and DATA states.
Example 5-2, Boundary-Scan Description Language (BSDL) for PQ2 Package Example (pg. 86)	New example.

Table 25. Data Sheet Version -003 to -004 Revision History (Sheet 2 of 2)

Section	Description
Section 4.6.1, AC Test Conditions (pg. 34)	Corrected reference to Figure 25.
Figure 22, Rise and Fall Time Derating at 85°C and Minimum VCC (pg. 41) Figure 23, ICC Active (Power Supply) vs. Frequency (pg. 41) Figure 24, ICC Active (Thermal) vs. Frequency (pg. 42) Figure 26, Output Delay vs. Temperature (pg. 43) Figure 27, Output Hold Times vs. Temperature (pg. 43) Figure 28, Output Delay vs. VCC (pg. 44)	New figures.
Figure 29, Cold Reset Waveform (pg. 45) Figure 30, Warm Reset Waveform (pg. 46)	Added values for RESET periods.
Figure 31, Entering ONCE Mode (pg. 47)	Added values for V_{CC} periods and indicated that CLKIN must be active.
Figures 33 through 39 (pages 49 through 55)	Revised DEN waveform.
Figures 38, 39, and 40 (pages 54 through 56)	Revised WAIT waveforms.
Figure 40, Burst, Non-Pipelined Read Request with Wait States, 8-Bit Bus (pg. 56)	Changed Burst and Non-Burst identifiers.
Figure 49, BOFF Functional Timing (pg. 65)	Moved Begin Request and End Request terminate outward.
Figure 58, Bus States (pg. 73)	Added READY signal to ADDRESS and DATA states.
Example 5-2, Boundary-Scan Description Language (BSDL) for PQ2 Package Example (pg. 86)	New example.

- — Nine Addressing Modes
- — User/Supervisor Protection Model

- Two-Way Set Associative Instruction Cache
  - — 80960JA - 2 Kbyte
  - — 80960JF - 4 Kbyte
  - — Programmable Cache Locking Mechanism
- Direct Mapped Data Cache
  - — 80960JA - 1 Kbyte
  - — 80960JF - 2 Kbyte
  - — Write Through Operation
- On-Chip Stack Frame Cache
  - — Seven Register Sets Can Be Saved
  - — Automatic Allocation on Call/Return
  - — 0-7 Frames Reserved for High-Priority Interrupts
- On-Chip Data RAM
  - — 1 Kbyte Critical Variable Storage
  - — Single-Cycle Access
- — Conditional Add, Subtract and Select
- — Processor Management

- High-Speed Interrupt Controller
  - — 31 Programmable Priorities
  - — Eight Maskable Pins plus NMI
  - — Up to 240 Vectors in Expanded Mode
- Two On-Chip Timers
  - — Independent 32-Bit Counting
  - — Clock Prescaling by 1, 2, 4 or 8
  - — Internal Interrupt Sources
- Halt Mode for Low Power
- IEEE 1149.1 (JTAG) Boundary Scan Compatibility
- Packages
  - — 132-Lead Pin Grid Array (PGA)
  - — 132-Lead Plastic Quad Flat Pack (PQFP)

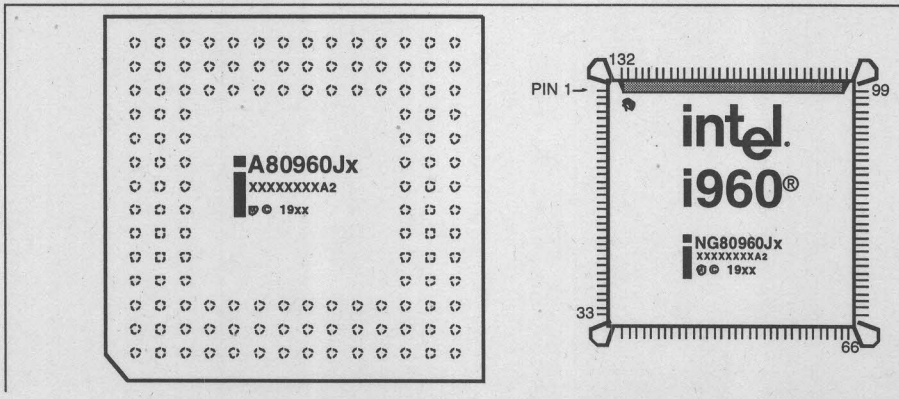


Figure 1. 80960JA/JF Microprocessors

Order Number: 272504-004

intel

PRELIMINARY

80960JA/JF

EMBEDDED 32-BIT MICROPROCESSOR

- Pin/Code Compatible with all 80960Jx Processors
- High-Performance Embedded Architecture
  - One Instruction/Clock Execution
  - Load/Store Programming Model
  - Sixteen 32-Bit Global Registers
  - Sixteen 32-Bit Local Registers (8 sets)
  - Nine Addressing Modes
  - User/Supervisor Protection Model
- Two-Way Set Associative Instruction Cache
  - 80960JA - 2 Kbyte
  - 80960JF - 4 Kbyte
  - Programmable Cache Locking Mechanism
- Direct Mapped Data Cache
  - 80960JA - 1 Kbyte
  - 80960JF - 2 Kbyte
  - Write Through Operation
- On-Chip Stack Frame Cache
  - Seven Register Sets Can Be Saved
  - Automatic Allocation on Call/Return
  - 0-7 Frames Reserved for High-Priority Interrupts
- On-Chip Data RAM
  - 1 Kbyte Critical Variable Storage
  - Single-Cycle Access
- High Bandwidth Burst Bus
  - 32-Bit Multiplexed Address/Data
  - Programmable Memory Configuration
  - Selectable 8-, 16-, 32-Bit Bus Widths
  - Supports Unaligned Accesses
  - Big or Little Endian Byte Ordering
- New Instructions
  - Conditional Add, Subtract and Select
  - Processor Management
- High-Speed Interrupt Controller
  - 31 Programmable Priorities
  - Eight Maskable Pins plus NMI
  - Up to 240 Vectors in Expanded Mode
- Two On-Chip Timers
  - Independent 32-Bit Counting
  - Clock Prescaling by 1, 2, 4 or 8
  - Internal Interrupt Sources
- Halt Mode for Low Power
- IEEE 1149.1 (JTAG) Boundary Scan Compatibility
- Packages
  - 132-Lead Pin Grid Array (PGA)
  - 132-Lead Plastic Quad Flat Pack (PQFP)

Figure 1. 80960JA/JF Microprocessors

Order Number 272564-004

1.0 PURPOSE

This document contains preliminary information for the 80960JA/JF microprocessor, including electrical characteristics and package pinout information. Detailed functional descriptions — other than parametric performance — are published in the *i960® Jx Microprocessor User's Guide* (272483).

Throughout this data sheet, references to "80960Jx" indicate features which apply to all of the following:

- 80960JA — 5V, 2 Kbyte instruction cache, 1 Kbyte data cache
- 80960JF — 5V, 4 Kbyte instruction cache, 2 Kbyte data cache
- 80960JD — 5V, 4 Kbyte instruction cache, 2 Kbyte data cache and clock doubling
- 80L960JA — 3.3 V version of the 80960JA
- 80L960JF — 3.3 V version of the 80960JF

2.0 80960JA/JF OVERVIEW

The 80960JA/JF offers high performance to cost-sensitive 32-bit embedded applications. The 80960JA/JF is object code compatible with the 80960 Core Architecture and is capable of sustained execution at the rate of one instruction per clock. This processor's features include generous instruction cache, data cache and data RAM. It also boasts a fast interrupt mechanism, dual programmable timer units and new instructions.

Memory subsystems for cost-sensitive embedded applications often impose substantial wait state penalties. The 80960JA/JF integrates considerable storage resources on-chip to decouple CPU execution from the external bus.

The 80960JA/JF rapidly allocates and deallocates local register sets during context switches. The processor needs to flush a register set to the stack only when it saves more than seven sets to its local register cache.

A 32-bit multiplexed burst bus provides a high-speed interface to system memory and I/O. A full complement of control signals simplifies the connection of the 80960JA/JF to external components. The user programs physical and logical memory attributes through memory-mapped control registers (MMRs) — an extension not found on the i960 Kx, Sx or Cx processors. Physical and logical configuration registers enable the processor to operate with all combinations of bus width and data object alignment. The processor supports a homogeneous byte ordering model.

This processor integrates two important peripherals: a timer unit and an interrupt controller. These and other hardware resources are programmed through memory-mapped control registers, an extension to the familiar 80960 architecture.

The timer unit (TU) offers two independent 32-bit timers for use as real-time system clocks and general-purpose system timing. These operate in either single-shot or auto-reload mode and can generate interrupts.

The interrupt controller unit (ICU) provides a flexible, low-latency means for requesting interrupts.The ICU provides full programmability of up to 240 interrupt sources into 31 priority levels. The ICU takes advantage of a cached priority table and optional routine caching to minimize interrupt latency. Local registers may be dedicated to high-priority interrupts to further reduce latency. Acting independently from the core, the ICU compares the priorities of posted interrupts with the current process priority, off-loading this task from the core. The ICU also supports the integrated timer interrupts.

The 80960JA/JF features a Halt mode designed to support applications where low power consumption is critical. The **halt** instruction shuts down instruction execution, resulting in a power savings of up to 90 percent.

The 80960JA/JF's testability features, including ONCE (On-Circuit Emulation) mode and Boundary Scan (JTAG), provide a powerful environment for design debug and fault diagnosis.

The *Solutions960®* program features a wide variety of development tools which support the i960 processor family. Many of these tools are developed by partner companies; some are developed by Intel, such as profile-driven optimizing compilers. For more information on these products, contact your local Intel representative.

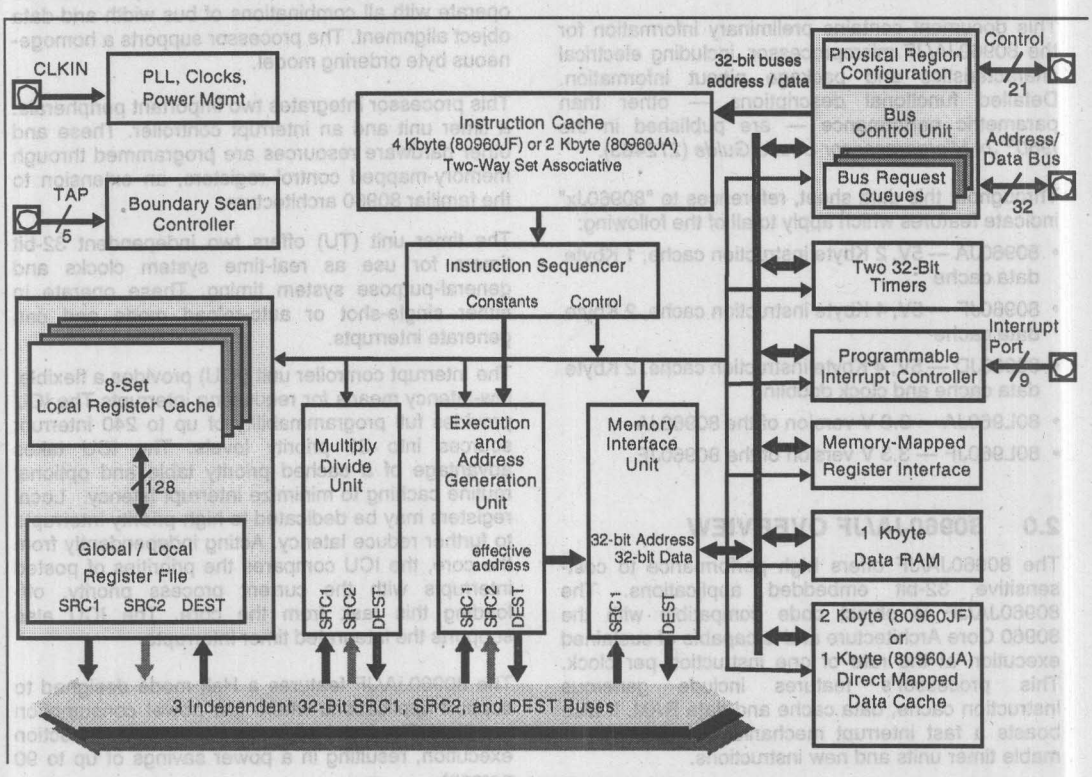


Figure 2. 80960JA/JF Block Diagram

2.1 80960 Processor Core

The 80960Jx family is a scalar implementation of the 80960 Core Architecture. Intel designed this processor core as a very high performance device that is also cost-effective. Factors that contribute to the core's performance include:

- Single-clock execution of most instructions
- Independent Multiply/Divide Unit
- Efficient instruction pipeline minimizes pipeline break latency
- Register and resource scoreboarding allow overlapped instruction execution
- 128-bit register bus speeds local register caching
- Two-way set associative, integrated instruction cache
- Direct-mapped, integrated data cache
- 1 Kbyte integrated data RAM delivers zero wait state program data

2.2 Burst Bus

A 32-bit high-performance bus controller interfaces the 80960JA/JF to external memory and peripherals. The BCU fetches instructions and transfers data at the rate of up to four 32-bit words per six clock cycles. The external address/data bus is multiplexed.

Users may configure the 80960JA/JF's bus controller to match an application's fundamental memory organization. Physical bus width is register-programmed for up to eight regions. Byte ordering and data caching are programmed through a group of logical memory templates and a defaults register.

The BCU's features include:

- Multiplexed external bus to minimize pin count
- 32-, 16- and 8-bit bus widths to simplify I/O interfaces
- External ready control for address-to-data, data-to-data and data-to-next-address wait state types
- Support for big or little endian byte ordering to facilitate the porting of existing program code
- Unaligned bus accesses performed transparently
- Three-deep load/store queue to decouple the bus from the core

Upon reset, the 80960JA/JF conducts an internal self test. Then, before executing its first instruction, it performs an external bus confidence test by performing a checksum on the first words of the initialization boot record (IBR).

The user may examine the contents of the caches at any time by executing special cache control instructions.

2.3 Timer Unit

The timer unit (TU) contains two independent 32-bit timers which are capable of counting at several clock rates and generating interrupts. Each is programmed by use of the TU registers. These memory-mapped registers are addressable on 32-bit boundaries. The timers have a single-shot mode and auto-reload capabilities for continuous operation. Each timer has an independent interrupt request to the interrupt controller. The TU can generate a fault when unauthorized writes from user mode are detected. Clock prescaling is supported.

2.4 Priority Interrupt Controller

A programmable interrupt controller manages up to 240 external sources through an 8-bit external interrupt port. Alternatively, the interrupt inputs may be configured for individual edge- or level-triggered inputs. The interrupt unit (IU) also accepts interrupts from the two on-chip timer channels and a single Non-Maskable Interrupt (NMI) pin. Interrupts are serviced according to their priority levels relative to the current process priority.

Low interrupt latency is critical to many embedded applications. As part of its highly flexible interrupt mechanism, the 80960JA/JF exploits several techniques to minimize latency:

- Interrupt vectors and interrupt handler routines can be reserved on-chip
- Register frames for high-priority interrupt handlers can be cached on-chip
- The interrupt stack can be placed in cacheable memory space

2.5 Instruction Set Summary

The 80960JA/JF adds several new instructions to the i960 core architecture. The new instructions are:

- Conditional Move
- Conditional Add
- Conditional Subtract
- Byte Swap
- Halt
- Cache Control
- Interrupt Control

Table 1 identifies the instructions that the 80960Jx supports. Refer to *i960® Jx Microprocessor User's Guide* (272483) for a detailed description of each instruction.

2.6 Faults and Debugging

The 80960Jx employs a comprehensive fault model. The processor responds to faults by making implicit calls to a fault handling routine. Specific information collected for each fault allows the fault handler to diagnose exceptions and recover appropriately.

The processor also has built-in debug capabilities. In software, the 80960Jx may be configured to detect as many as seven different trace event types. Alternatively, **mark** and **fmark** instructions can generate

micron manufacturing process. The processor's sub-micron topology provides the circuit density for optimal cache size and high operating speeds while dissipating modest power. The processor also uses dynamic power management to turn off clocks to unused circuits.

Users may program the 80960Jx to enter Halt mode for maximum power savings. In Halt mode, the processor core stops completely while the integrated peripherals continue to function, reducing overall power requirements up to 90 percent. Processor execution resumes from internally or externally generated interrupts.

2.8 Test Features

The 80960Jx incorporates numerous features which enhance the user's ability to test both the processor and the system to which it is attached. These features include ONCE (On-Circuit Emulation) mode and Boundary Scan (JTAG).

The 80960Jx provides testability features compatible with IEEE Standard Test Access Port and Boundary Scan Architecture (IEEE Std. 1149.1).

One of the boundary scan instructions, **HIGHZ**, forces the processor to float all its output pins (ONCE mode). ONCE mode can also be initiated at reset without using the boundary scan mechanism.

ONCE mode is useful for board-level testing. This feature allows a mounted 80960JA/JF to electrically "remove" itself from a circuit board. This allows for system-level testing where a remote tester — such as an in-circuit emulator — can exercise the processor system.

The provided test logic does not interfere with component or circuit board behavior and ensures that components function correctly, connections

2.9 Memory-Mapped Control Registers

The 80960JA/JF, though compliant with i960 series processor core, has the added advantage of memory-mapped, internal control registers not found on the i960 Kx, Sx or Cx processors. These give software the interface to easily read and modify internal control registers.

Each of these registers is accessed as a memory-mapped, 32-bit register. Access is accomplished through regular memory-format instructions. The processor ensures that these accesses do not generate external bus cycles.

2.10 Data Types and Memory Addressing Modes

As with all i960 family processors, the 80960JA/JF instruction set supports several data types and formats:

- Bit
- Bit fields
- Integer (8-, 16-, 32-, 64-bit)
- Ordinal (8-, 16-, 32-, 64-bit unsigned integers)
- Triple word (96 bits)
- Quad word (128 bits)

The 80960JA/JF provides a full set of addressing modes for C and assembly programming:

- Two Absolute modes
- Five Register Indirect modes
- Index with displacement
- IP with displacement

Table 1. 80960Jx Instruction Set

Data Movement	Arithmetic	Logical	Bit, Bit Field and Byte
Load	Add	And	Set Bit
Store	Subtract	Not And	Clear Bit
Move	Multiply	And Not	Not Bit
*Conditional Select	Divide	Or	Alter Bit
Load Address	Remainder	Exclusive Or	Scan For Bit
	Modulo	Not Or	Span Over Bit
	Shift	Or Not	Extract
	Extended Shift	Nor	Modify
	Extended Multiply	Exclusive Nor	Scan Byte for Equal
	Extended Divide	Not	*Byte Swap
	Add with Carry	Nand	
	Subtract with Carry		
	*Conditional Add		
	*Conditional Subtract		
	Rotate		
Comparison	**Branch**	**Call/Return**	**Fault**
Compare	Unconditional Branch	Call	Conditional Fault
Conditional Compare	Conditional Branch	Call Extended	Synchronize Faults
Compare and Increment	Compare and Branch	Call System	
Compare and Decrement		Return	
Test Condition Code		Branch and Link	
Check Bit			
Debug	**Processor Management**	**Atomic**	
Modify Trace Controls	Flush Local Registers	Atomic Add	
Mark	Modify Arithmetic Controls	Atomic Modify	
Force Mark	Modify Process Controls		
	*Halt		
	System Control		
	*Cache Control		
	*Interrupt Control		

NOTE: Asterisk (*) denotes new 80960Jx instructions unavailable on 80960CA/CF, 80960KA/KB and 80960SA/SB implementations.

3.0 PACKAGE INFORMATION

The 80960JA/JF will be offered in several speed and package types. The 132-pin Pin Grid Array (PGA) device will be specified for operation at V_{CC} = 5.0 V ± 5% over a case temperature range of 0° to 100°C:

- A80960JA/JF-33 (33 MHz)
- A80960JA/JF-25 (25 MHz)
- A80960JA/JF-16 (16 MHz)

The 132-pin Plastic Quad Flatpack (PQFP) devices will be specified for operation at V_{CC} = 5.0 V ± 5% over a case temperature range of 0° to 100°C:

- NG80960JA/JF-33 (33 MHz)
- NG80960JA/JF-25 (25 MHz)
- NG80960JA/JF-16 (16 MHz)

For complete package specifications and information, refer to Intel's Packaging Handbook (240800).

3.1 Pin Descriptions

This section describes the pins for the 80960JA/JF in the 132-pin ceramic Pin Grid Array (PGA) package and 132-lead Plastic Quad Flatpack Package (PQFP).

Section 3.1.1, Functional Pin Definitions describes pin function; **Section 3.1.2, 80960Jx 132-Lead PGA Pinout** and **Section 3.1.3, 80960Jx PQFP Pinout** define the signal and pin locations for the supported package types.

3.1.1 Functional Pin Definitions

Table 2 presents the legend for interpreting the pin descriptions which follow. Pins associated with the bus interface are described in Table 3. Pins associated with basic control and test functions are described in Table 4. Pins associated with the Interrupt Unit are described in Table 5.

Table 2. Pin Description Nomenclature

Symbol	Description
I	Input pin only.
O	Output pin only.
I/O	Pin can be either an input or output.
–	Pin must be connected as described.
S	Synchronous. Inputs must meet setup and hold times relative to CLKIN for proper operation. S(E) Edge sensitive input S(L) Level sensitive input
A (...)	Asynchronous. Inputs may be asynchronous relative to CLKIN. A(E) Edge sensitive input A(L) Level sensitive input
R (...)	While the processor's $\overline{\text{RESET}}$ pin is asserted, the pin: R(1) is driven to V_{CC} R(0) is driven to V_{SS} R(Q) is a valid output R(X) is driven to unknown state R(H) is pulled up to V_{CC}
H (...)	While the processor is in the hold state, the pin: H(1) is driven to V_{CC} H(0) is driven to V_{SS} H(Q) Maintains previous state or continues to be a valid output H(Z) Floats
P (...)	While the processor is halted, the pin: P(1) is driven to V_{CC} P(0) is driven to V_{SS} P(Q) Maintains previous state or continues to be a valid output

Table 3. Pin Description — External Bus Signals (Sheet 1 of 4)

NAME	TYPE	DESCRIPTION
AD31:0	**I/O** S(L) R(X) H(Z) P(Q)	**ADDRESS / DATA BUS** carries 32-bit physical addresses and 8-, 16- or 32-bit data to and from memory. During an address (T_a) cycle, bits 31:2 contain a physical word address (bits 0-1 indicate SIZE; see below). During a data (T_d) cycle, read or write data is present on one or more contiguous bytes, comprising AD31:24, AD23:16, AD15:8 and AD7:0. During write operations, unused pins are driven to determinate values. SIZE, which comprises bits 0-1 of the AD lines during a T_a cycle, specifies the number of data transfers during the bus transaction. AD1 AD0 Bus Transfers 0 0 1 Transfer 0 1 2 Transfers 1 0 3 Transfers 1 1 4 Transfers When the processor enters Halt mode, if the previous bus operation was a: • write — AD31:2 are driven with the last data value on the AD bus. • read — AD31:4 are driven with the last address value on the AD bus; AD3:2 are driven with the value of A3:2 from the last data cycle. Typically, AD1:0 reflect the SIZE information of the last bus transaction (either instruction fetch or load/store) that was executed before entering Halt mode.
ALE	**O** R(0) H(Z) P(0)	**ADDRESS LATCH ENABLE** indicates the transfer of a physical address. ALE is asserted during a T_a cycle and deasserted before the beginning of the T_d state. It is active HIGH and floats to a high impedance state during a hold cycle (T_h).
$\overline{\text{ALE}}$	**O** R(1) H(Z) P(1)	**ADDRESS LATCH ENABLE** indicates the transfer of a physical address. $\overline{\text{ALE}}$ is the inverted version of ALE. This signal gives the 80960JA/JF a high degree of compatibility with existing 80960Kx systems.
$\overline{\text{ADS}}$	**O** R(1) H(Z) P(1)	**ADDRESS STROBE** indicates a valid address and the start of a new bus access. The processor asserts $\overline{\text{ADS}}$ for the entire T_a cycle. External bus control logic typically samples $\overline{\text{ADS}}$ at the end of the cycle.
A3:2	**O** R(X) H(Z) P(Q)	**ADDRESS3:2** comprise a partial demultiplexed address bus. *32-bit memory accesses:* the processor asserts address bits A3:2 during T_a. The partial word address increments with each assertion of RDYRCV during a burst. *16-bit memory accesses:* the processor asserts address bits A3:1 during T_a with A1 driven on the $\overline{\text{BE1}}$ pin. The partial short word address increments with each assertion of RDYRCV during a burst. *8-bit memory accesses:* the processor asserts address bits A3:0 during T_a, with A1:0 driven on $\overline{\text{BE1:0}}$. The partial byte address increments with each assertion of RDYRCV during a burst.

		$\overline{BE0}$ enables data on AD7:0 *16-bit bus:* $\overline{BE3}$ becomes Byte High Enable (enables data on AD15:8) $\overline{BE2}$ is not used (state is high) $\overline{BE1}$ becomes Address Bit 1 (A1) $\overline{BE0}$ becomes Byte Low Enable (enables data on AD7:0) *8-bit bus:* $\overline{BE3}$ is not used (state is high) $\overline{BE2}$ is not used (state is high) $\overline{BE1}$ becomes Address Bit 1 (A1) $\overline{BE0}$ becomes Address Bit 0 (A0) The processor asserts byte enables, byte high enable and byte low enable during T_a. Since unaligned bus requests are split into separate bus transactions, these signals do not toggle during a burst. They remain active through the last T_d cycle. For accesses to 8- and 16-bit memory, the processor asserts the address bits in conjunction with A3:2 described above.
WIDTH/ HLTD1:0	**O** R(0) H(Z) P(1)	**WIDTH/HALTED** signals denote the physical memory attributes for a bus transaction: WIDTH/HLTD1 WIDTH/HLTD0 0 0 8 Bits Wide 0 1 16 Bits Wide 1 0 32 Bits Wide 1 1 Processor Halted The processor floats the WIDTH/HLTD pins whenever it relinquishes the bus in response to a HOLD request, regardless of prior operating state.
D/$\overline{C}$	**O** P(Ú)	**DATA/CODE** indicates that a bus access is a data access (1) or an instruction 1 = data access
W/$\overline{R}$	**O** R(0) H(Z) P(Q)	**WRITE/READ** specifies, during a T_a cycle, whether the operation is a write (1) or read (0). It is latched on-chip and remains valid during T_d cycles. 0 = read 1 = write

Table 3. Pin Description — External Bus Signals (Sheet 3 of 4)

NAME	TYPE	DESCRIPTION
DT/$\overline{R}$	O R(0) H(Z) P(Q)	**DATA TRANSMIT / RECEIVE** indicates the direction of data transfer to and from the address/data bus. It is low during T_a and T_w/T_d cycles for a read; it is high during T_a and T_w/T_d cycles for a write. DT/$\overline{R}$ never changes state when $\overline{DEN}$ is asserted. 0 = receive 1 = transmit
$\overline{DEN}$	O R(1) H(Z) P(1)	**DATA ENABLE** indicates data transfer cycles during a bus access. $\overline{DEN}$ is asserted at the start of the first data cycle in a bus access and deasserted at the end of the last data cycle. $\overline{DEN}$ is used with DT/$\overline{R}$ to provide control for data transceivers connected to the data bus. 0 = data cycle 1 = not data cycle
$\overline{BLAST}$	O R(1) H(Z) P(1)	**BURST LAST** indicates the last transfer in a bus access. $\overline{BLAST}$ is asserted in the last data transfer of burst and non-burst accesses. $\overline{BLAST}$ remains active as long as wait states are inserted via the $\overline{RDYRCV}$ pin. $\overline{BLAST}$ becomes inactive after the final data transfer in a bus cycle. 0 = last data transfer 1 = not last data transfer
$\overline{RDYRCV}$	I S(L)	**READY/RECOVER** indicates that data on AD lines can be sampled or removed. When $\overline{RDYRCV}$ is not asserted during a T_d cycle, the T_d cycle is extended to the next cycle by inserting a wait state (T_w). 0 = sample data 1 = don't sample data The $\overline{RDYRCV}$ pin has another function during the recovery (T_r) state. The processor continues to insert additional recovery states until it samples the pin HIGH. This function gives slow external devices sufficient time to float their buffers before the processor begins to drive address again. 0 = insert wait states 1 = recovery complete
$\overline{LOCK}$/ ONCE	I/O S(L) R(H) H(Z) P(1)	**BUS LOCK** indicates that an atomic read-modify-write operation is in progress. The $\overline{LOCK}$ output is asserted in the first clock of an atomic operation and deasserted in the last data transfer of the sequence. The processor does not grant HOLDA while it is asserting $\overline{LOCK}$. This prevents external agents from accessing memory involved in semaphore operations. 0 = Atomic read-modify-write in progress 1 = Atomic read-modify-write not in progress **ONCE MODE:** The processor samples the $\overline{ONCE}$ input during reset. When it is asserted LOW at the end of reset, the processor enters ONCE mode. In ONCE mode, the processor stops all clocks and floats all output pins. The pin has a weak internal pullup which is active during reset to ensure normal operation when the pin is left unconnected. 0 = ONCE mode enabled 1 = ONCE mode not enabled

Table 3. Pin Description — External Bus Signals (Sheet 4 of 4)

NAME	TYPE	DESCRIPTION
HOLD	I S(L)	**HOLD:** A request from an external bus master to acquire the bus. When the processor receives HOLD and grants bus control to another master, it asserts HOLDA, floats the address/data and control lines and enters the T_h state. When HOLD is deasserted, the processor deasserts HOLDA and enters either the T_i or T_a state, resuming control of the address/data and control lines. 0 = no hold request 1 = hold request
HOLDA	O R(Q) H(1) P(Q)	**HOLD ACKNOWLEDGE** indicates to an external bus master that the processor has relinquished control of the bus. The processor can grant HOLD requests and enter the T_h state during reset and while halted as well as during regular operation. 0 = hold not acknowledged 1 = hold acknowledged
BSTAT	O R(0) H(Q) P(0)	**BUS STATUS** indicates that the processor may soon stall unless it has sufficient access to the bus; see *i960® Jx Microprocessor User's Guide* (272483). Arbitration logic can examine this signal to determine when an external bus master should acquire/relinquish the bus. 0 = no potential stall 1 = potential stall

Table 4. Pin Description — Processor Control Signals, Test Signals and Power (Sheet 1 of 2)

NAME	TYPE	DESCRIPTION
CLKIN	I	**CLOCK INPUT** provides the processor's fundamental time base; both the processor core and the external bus run at the CLKIN rate. All input and output timings are specified relative to a rising CLKIN edge.
$\overline{\text{RESET}}$	I A(L)	**RESET** initializes the processor and clears its internal logic. During reset, the processor places the address/data bus and control output pins in their idle (inactive) states. During reset, the input pins are ignored with the exception of $\overline{\text{LOCK}}$/$\overline{\text{ONCE}}$, STEST and HOLD. The $\overline{\text{RESET}}$ pin has an internal synchronizer. To ensure predictable processor initialization during power up, $\overline{\text{RESET}}$ must be asserted a minimum of 10,000 CLKIN cycles with V_{CC} and CLKIN stable. On a warm reset, $\overline{\text{RESET}}$ should be asserted for a minimum of 15 cycles.
STEST	I S(L)	**SELF TEST** enables or disables the processor's internal self-test feature at initialization. STEST is examined at the end of reset. When STEST is asserted, the processor performs its internal self-test and the external bus confidence test. When STEST is deasserted, the processor performs only the external bus confidence test. 0 = self test disabled 1 = self test enabled

Table 4. Pin Description — Processor Control Signals, Test Signals and Power (Sheet 2 of 2)

NAME	TYPE	DESCRIPTION
$\overline{\text{FAIL}}$	O R(0) H(Q) P(1)	**FAIL** indicates a failure of the processor's built-in self-test performed during initialization. $\overline{\text{FAIL}}$ is asserted immediately upon reset and toggles during self-test to indicate the status of individual tests: • When self-test passes, the processor deasserts $\overline{\text{FAIL}}$ and begins operation from user code. • When self-test fails, the processor asserts $\overline{\text{FAIL}}$ and then stops executing. 0 = self test failed 1 = self test passed
TCK	I	**TEST CLOCK** is a CPU input which provides the clocking function for IEEE 1149.1 Boundary Scan Testing (JTAG). State information and data are clocked into the processor on the rising edge; data is clocked out of the processor on the falling edge.
TDI	I S(L)	**TEST DATA INPUT** is the serial input pin for JTAG. TDI is sampled on the rising edge of TCK, during the SHIFT-IR and SHIFT-DR states of the Test Access Port.
TDO	O R(Q) HQ) P(Q)	**TEST DATA OUTPUT** is the serial output pin for JTAG. TDO is driven on the falling edge of TCK during the SHIFT-IR and SHIFT-DR states of the Test Access Port. At other times, TDO floats. TDO does not float during ONCE mode.
$\overline{\text{TRST}}$	I A(L)	**TEST RESET** asynchronously resets the Test Access Port (TAP) controller function of IEEE 1149.1 Boundary Scan testing (JTAG). When using the Boundary Scan feature, connect a pulldown resistor between this pin and V_{SS}. When TAP is not used, this pin must be connected to V_{SS}; however, no resistor is required. See Section 4.3, Connection Recommendations (pg. 22).
TMS	I S(L)	**TEST MODE SELECT** is sampled at the rising edge of TCK to select the operation of the test logic for IEEE 1149.1 Boundary Scan testing.
V_{CC}	–	**POWER** pins intended for external connection to a V_{CC} board plane.
V_{CCPLL}	–	**PLL POWER** is a separate V_{CC} supply pin for the phase lock loop clock generator. It is intended for external connection to the V_{CC} board plane. In noisy environments, add a simple bypass filter circuit to reduce noise-induced clock jitter and its effects on timing relationships.
V_{SS}	–	**GROUND** pins intended for external connection to a V_{SS} board plane.
NC	–	**NO CONNECT** pins. Do not make any system connections to these pins.

		Mixed Mode: The XINT7:5 pins act as dedicated sources and the XINT4:0 pins act as the five most significant bits of a vectored source. The least significant bits of the vectored source are set to 010_2 internally. Unused external interrupt pins should be connected to V_{CC}.
NMI	I A(E)	**NON-MASKABLE INTERRUPT** causes a non-maskable interrupt event to occur. NMI is the highest priority interrupt source and is falling edge-triggered. If NMI is unused, it should be connected to V_{CC}.

3.1.2 80960Jx 132-Lead PGA Pinout

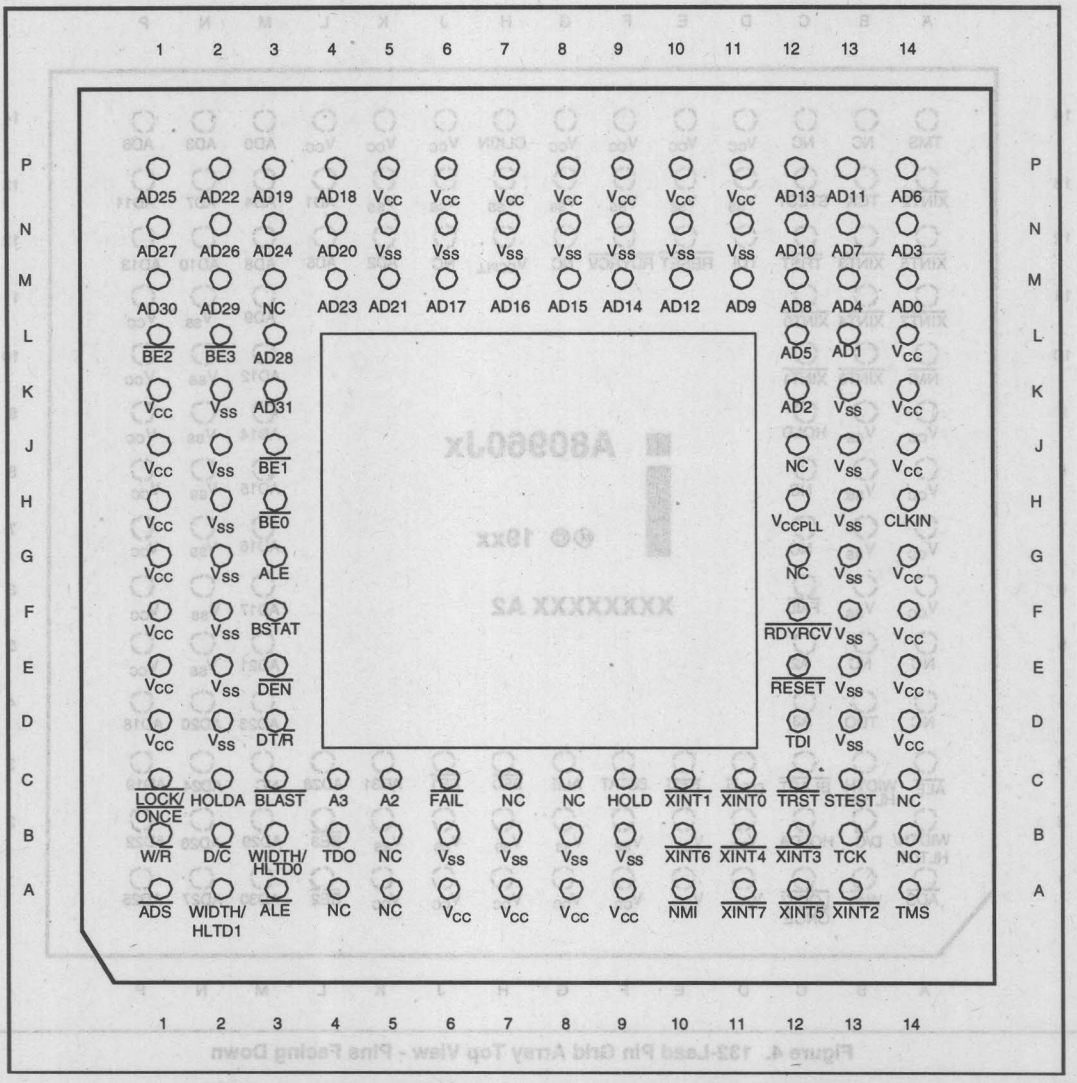


Figure 3. 132-Lead Pin Grid Array Bottom View - Pins Facing Up

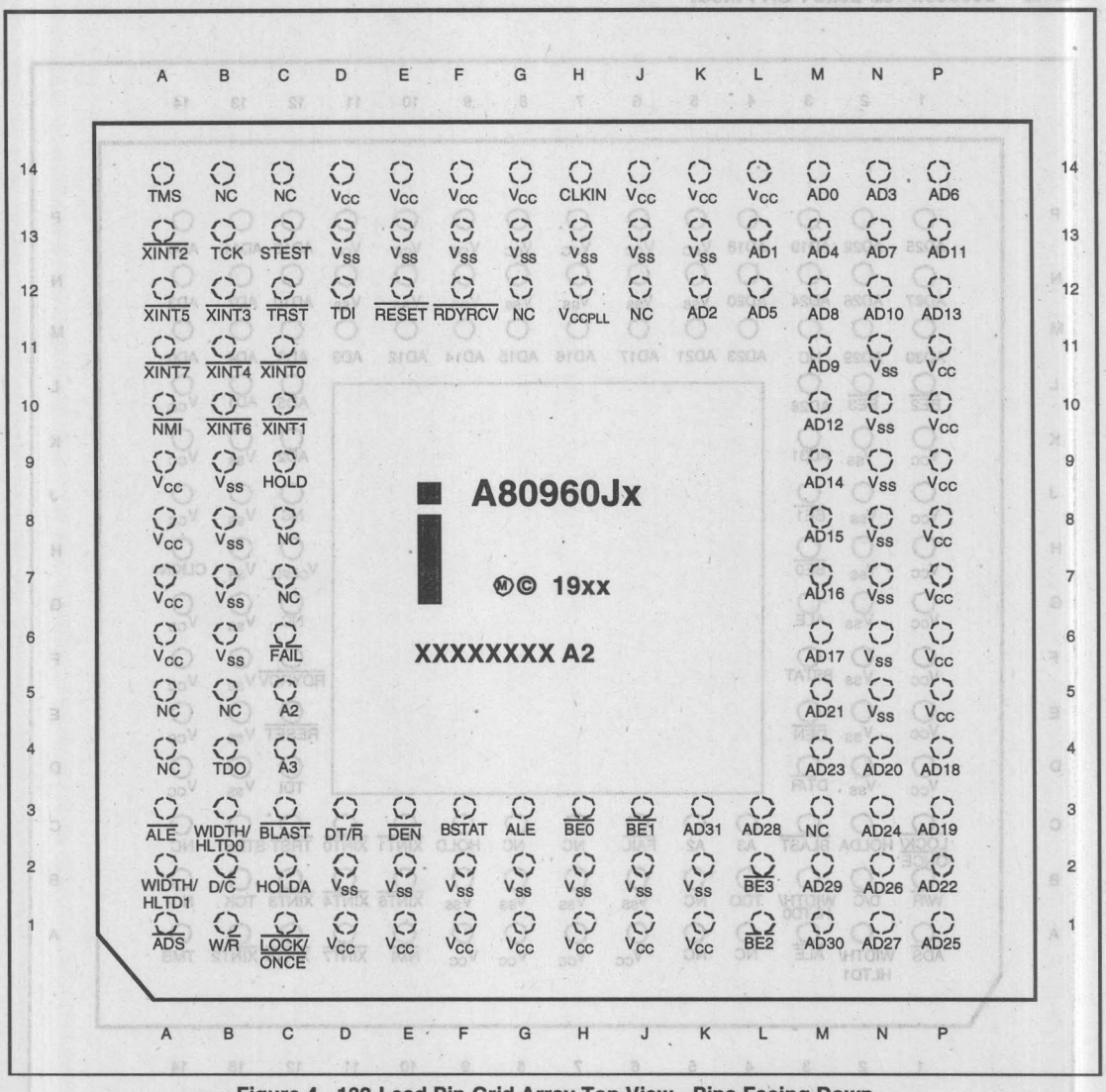


Figure 4. 132-Lead Pin Grid Array Top View - Pins Facing Down

AD3	N14	BE1	J3	V_{CC}	A7	V_{SS}	F2
AD4	M13	$\overline{BE2}$	L1	V_{CC}	A8	V_{SS}	F13
AD5	L12	$\overline{BE3}$	L2	V_{CC}	A9	V_{SS}	G2
AD6	P14	$\overline{BLAST}$	C3	V_{CC}	D1	V_{SS}	G13
AD7	N13	BSTAT	F3	V_{CC}	D14	V_{SS}	H2
AD8	M12	CLKIN	H14	V_{CC}	E1	V_{SS}	H13
AD9	M11	D/$\overline{C}$	B2	V_{CC}	E14	V_{SS}	J2
AD10	N12	$\overline{DEN}$	E3	V_{CC}	F1	V_{SS}	J13
AD11	P13	DT/$\overline{R}$	D3	V_{CC}	F14	V_{SS}	K2
AD12	M10	$\overline{FAIL}$	C6	V_{CC}	G1	V_{SS}	K13
AD13	P12	HOLD	C9	V_{CC}	G14	V_{SS}	N5
AD14	M9	HOLDA	C2	V_{CC}	H1	V_{SS}	N6
AD15	M8	$\overline{LOCK/ONCE}$	C1	V_{CC}	J1	V_{SS}	N7
AD16	M7	NC	A4	V_{CC}	J14	V_{SS}	N8
AD17	M6	NC	A5	V_{CC}	K1	V_{SS}	N9
AD18	P4	NC	B5	V_{CC}	K14	V_{SS}	N10
AD19	P3	NC	B14	V_{CC}	L14	V_{SS}	N11
AD20	N4	NC	C7	V_{CC}	P5	W/$\overline{R}$	B1
AD21	M5	NC	C8	V_{CC}	P6	WIDTH/HLTD0	B3
AD22	P2	NC	C14	V_{CC}	P7	WIDTH/HLTD1	A2
AD23	M4	NC	G12	V_{CC}	P8	$\overline{XINT0}$	C11
AD24	N3	NC	J12	V_{CC}	P9	$\overline{XINT1}$	C10
AD25	P1	NC	M3	V_{CC}	P10	$\overline{XINT2}$	A13
AD26	N2	$\overline{NMI}$	A10	V_{CC}	P11	$\overline{XINT3}$	B12
AD27	N1	$\overline{RDYRCV}$	F12	V_{CCPLL}	H12	$\overline{XINT4}$	B11
AD28	L3	$\overline{RESET}$	E12	V_{SS}	B6	$\overline{XINT5}$	A12
AD29	M2	STEST	C13	V_{SS}	B7	$\overline{XINT6}$	B10
AD30	M1	TCK	B13	V_{SS}	B8	$\overline{XINT7}$	A11

NOTE: Do not connect any external logic to pins marked NC (no connect pins).

A8	V_{CC}	C13	STEST	J2	V_{SS}	N3	AD24
A9	V_{CC}	C14	NC	J3	$\overline{BE1}$	N4	AD20
A10	$\overline{NMI}$	D1	V_{CC}	J12	NC	N5	V_{SS}
A11	$\overline{XINT7}$	D2	V_{SS}	J13	V_{SS}	N6	V_{SS}
A12	$\overline{XINT5}$	D3	DT/$\overline{R}$	J14	V_{CC}	N7	V_{SS}
A13	$\overline{XINT2}$	D12	TDI	K1	V_{CC}	N8	V_{SS}
A14	TMS	D13	V_{SS}	K2	V_{SS}	N9	V_{SS}
B1	W/$\overline{R}$	D14	V_{CC}	K3	AD31	N10	V_{SS}
B2	D/$\overline{C}$	E1	V_{CC}	K12	AD2	N11	V_{SS}
B3	WIDTH/HLTD0	E2	V_{SS}	K13	V_{SS}	N12	AD10
B4	TDO	E3	$\overline{DEN}$	K14	V_{CC}	N13	AD7
B5	NC	E12	$\overline{RESET}$	L1	$\overline{BE2}$	N14	AD3
B6	V_{SS}	E13	V_{SS}	L2	$\overline{BE3}$	P1	AD25
B7	V_{SS}	E14	V_{CC}	L3	AD28	P2	AD22
B8	V_{SS}	F1	V_{CC}	L12	AD5	P3	AD19
B9	V_{SS}	F2	V_{SS}	L13	AD1	P4	AD18
B10	$\overline{XINT6}$	F3	BSTAT	L14	V_{CC}	P5	V_{CC}
B11	$\overline{XINT4}$	F12	$\overline{RDYRCV}$	M1	AD30	P6	V_{CC}
B12	$\overline{XINT3}$	F13	V_{SS}	M2	AD29	P7	V_{CC}
B13	TCK	F14	V_{CC}	M3	NC	P8	V_{CC}
B14	NC	G1	V_{CC}	M4	AD23	P9	V_{CC}
C1	$\overline{LOCK/ONCE}$	G2	V_{SS}	M5	AD21	P10	V_{CC}
C2	HOLDA	G3	ALE	M6	AD17	P11	V_{CC}
C3	$\overline{BLAST}$	G12	NC	M7	AD16	P12	AD13
C4	A3	G13	V_{SS}	M8	AD15	P13	AD11
C5	A2	G14	V_{CC}	M9	AD14	P14	AD6

NOTE: Do not connect any external logic to pins marked NC (no connect pins).

3.1.3 80960Jx PQFP Pinout

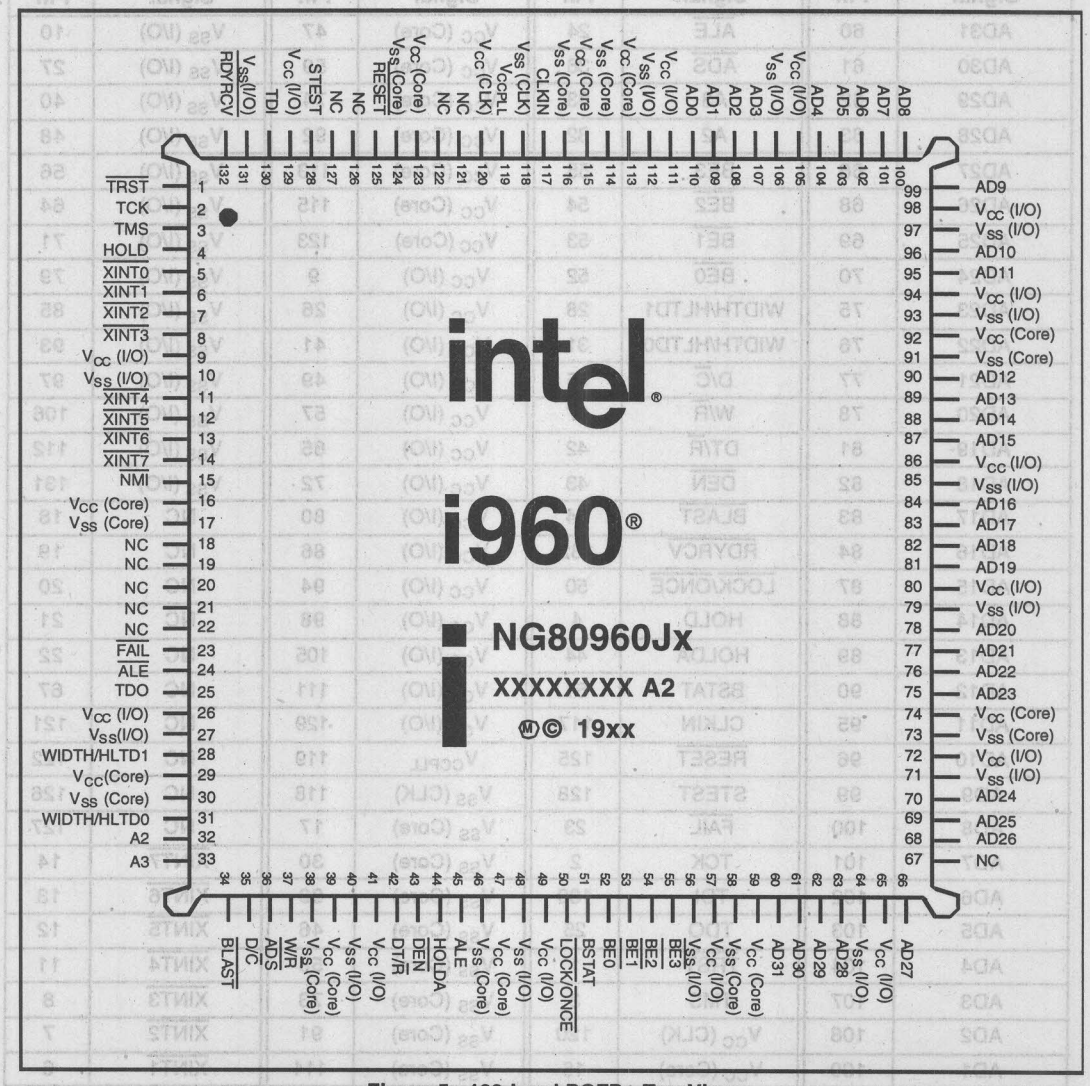


Figure 5. 132-Lead PQFP - Top View

Table 8. 132-Lead PQFP Pinout — In Signal Order

Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin
AD31	60	$\overline{ALE}$	24	V_{CC} (Core)	47	V_{SS} (I/O)	10
AD30	61	$\overline{ADS}$	36	V_{CC} (Core)	59	V_{SS} (I/O)	27
AD29	62	A3	33	V_{CC} (Core)	74	V_{SS} (I/O)	40
AD28	63	A2	32	V_{CC} (Core)	92	V_{SS} (I/O)	48
AD27	66	$\overline{BE3}$	55	V_{CC} (Core)	113	V_{SS} (I/O)	56
AD26	68	$\overline{BE2}$	54	V_{CC} (Core)	115	V_{SS} (I/O)	64
AD25	69	$\overline{BE1}$	53	V_{CC} (Core)	123	V_{SS} (I/O)	71
AD24	70	$\overline{BE0}$	52	V_{CC} (I/O)	9	V_{SS} (I/O)	79
AD23	75	WIDTH/HLTD1	28	V_{CC} (I/O)	26	V_{SS} (I/O)	85
AD22	76	WIDTH/HLTD0	31	V_{CC} (I/O)	41	V_{SS} (I/O)	93
AD21	77	D/$\overline{C}$	35	V_{CC} (I/O)	49	V_{SS} (I/O)	97
AD20	78	W/$\overline{R}$	37	V_{CC} (I/O)	57	V_{SS} (I/O)	106
AD19	81	DT/$\overline{R}$	42	V_{CC} (I/O)	65	V_{SS} (I/O)	112
AD18	82	$\overline{DEN}$	43	V_{CC} (I/O)	72	V_{SS} (I/O)	131
AD17	83	$\overline{BLAST}$	34	V_{CC} (I/O)	80	NC	18
AD16	84	$\overline{RDYRCV}$	132	V_{CC} (I/O)	86	NC	19
AD15	87	$\overline{LOCK/ONCE}$	50	V_{CC} (I/O)	94	NC	20
AD14	88	HOLD	4	V_{CC} (I/O)	98	NC	21
AD13	89	HOLDA	44	V_{CC} (I/O)	105	NC	22
AD12	90	BSTAT	51	V_{CC} (I/O)	111	NC	67
AD11	95	CLKIN	117	V_{CC} (I/O)	129	NC	121
AD10	96	$\overline{RESET}$	125	V_{CCPLL}	119	NC	122
AD9	99	STEST	128	V_{SS} (CLK)	118	NC	126
AD8	100	$\overline{FAIL}$	23	V_{SS} (Core)	17	NC	127
AD7	101	TCK	2	V_{SS} (Core)	30	$\overline{XINT7}$	14
AD6	102	TDI	130	V_{SS} (Core)	38	$\overline{XINT6}$	13
AD5	103	TDO	25	V_{SS} (Core)	46	$\overline{XINT5}$	12
AD4	104	$\overline{TRST}$	1	V_{SS} (Core)	58	$\overline{XINT4}$	11
AD3	107	TMS	3	V_{SS} (Core)	73	$\overline{XINT3}$	8
AD2	108	V_{CC} (CLK)	120	V_{SS} (Core)	91	$\overline{XINT2}$	7
AD1	109	V_{CC} (Core)	16	V_{SS} (Core)	114	$\overline{XINT1}$	6
AD0	110	V_{CC} (Core)	29	V_{SS} (Core)	116	$\overline{XINT0}$	5
ALE	45	V_{CC} (Core)	39	V_{SS} (Core)	124	$\overline{NMI}$	15

NOTE: Do not connect any external logic to pins marked NC (no connect pins).

Table 9. 132-Lead PQFP Pinout — In Pin Order

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
1	$\overline{TRST}$	34	$\overline{BLAST}$	67	NC	100	AD8
2	TCK	35	D/$\overline{C}$	68	AD26	101	AD7
3	TMS	36	$\overline{ADS}$	69	AD25	102	AD6
4	HOLD	37	W/$\overline{R}$	70	AD24	103	AD5
5	$\overline{XINT0}$	38	V_{SS} (Core)	71	V_{SS} (I/O)	104	AD4
6	$\overline{XINT1}$	39	V_{CC} (Core)	72	V_{CC} (I/O)	105	V_{CC} (I/O)
7	$\overline{XINT2}$	40	V_{SS} (I/O)	73	V_{SS} (Core)	106	V_{SS} (I/O)
8	$\overline{XINT3}$	41	V_{CC} (I/O)	74	V_{CC} (Core)	107	AD3
9	V_{CC} (I/O)	42	DT/$\overline{R}$	75	AD23	108	AD2
10	V_{SS} (I/O)	43	$\overline{DEN}$	76	AD22	109	AD1
11	$\overline{XINT4}$	44	HOLDA	77	AD21	110	AD0
12	$\overline{XINT5}$	45	ALE	78	AD20	111	V_{CC} (I/O)
13	$\overline{XINT6}$	46	V_{SS} (Core)	79	V_{SS} (I/O)	112	V_{SS} (I/O)
14	$\overline{XINT7}$	47	V_{CC} (Core)	80	V_{CC} (I/O)	113	V_{CC} (Core)
15	$\overline{NMI}$	48	V_{SS} (I/O)	81	AD19	114	V_{SS} (Core)
16	V_{CC} (Core)	49	V_{CC} (I/O)	82	AD18	115	V_{CC} (Core)
17	V_{SS} (Core)	50	$\overline{LOCK}$/$\overline{ONCE}$	83	AD17	116	V_{SS} (Core)
18	NC	51	BSTAT	84	AD16	117	CLKIN
19	NC	52	$\overline{BE0}$	85	V_{SS} (I/O)	118	V_{SS} (CLK)
20	NC	53	$\overline{BE1}$	86	V_{CC} (I/O)	119	V_{CCPLL}
21	NC	54	$\overline{BE2}$	87	AD15	120	$V_{CC\ (CLK)}$
22	NC	55	$\overline{BE3}$	88	AD14	121	NC
23	$\overline{FAIL}$	56	V_{SS} (I/O)	89	AD13	122	NC
24	$\overline{ALE}$	57	V_{CC} (I/O)	90	AD12	123	V_{CC} (Core)
25	TDO	58	V_{SS} (Core)	91	V_{SS} (Core)	124	V_{SS} (Core)
26	V_{CC} (I/O)	59	V_{CC} (Core)	92	V_{CC} (Core)	125	$\overline{RESET}$
27	V_{SS} (I/O)	60	AD31	93	V_{SS} (I/O)	126	NC
28	WIDTH/HLTD1	61	AD30	94	V_{CC} (I/O)	127	NC
29	V_{CC} (Core)	62	AD29	95	AD11	128	STEST
30	V_{SS} (Core)	63	AD28	96	AD10	129	V_{CC} (I/O)
31	WIDTH/HLTD0	64	V_{SS} (I/O)	97	V_{SS} (I/O)	130	TDI
32	A2	65	V_{CC} (I/O)	98	V_{CC} (I/O)	131	V_{SS} (I/O)
33	A3	66	AD27	99	AD9	132	$\overline{RDYRCV}$

NOTE: Do not connect any external logic to pins marked NC (no connect pins).

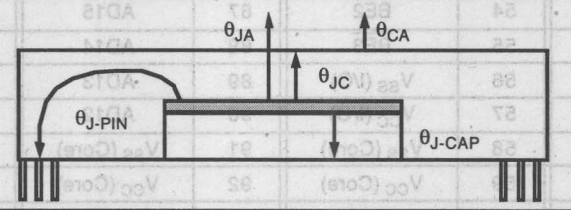
θ_{CA} is the thermal resistance from case to ambient. Use the following equation to calculate T_A, the maximum ambient temperature to conform to a particular case temperature:

$$T_A = T_C - P\ (\theta_{CA})$$

Compute P by multiplying I_{CC} from Table 14 and V_{CC}. Values for θ_{JC} and θ_{CA} are given in Table 10 for the PGA package and Table 11 for the PQFP package. For high speed operation, the processor's θ_{JA} may be significantly reduced by adding a heatsink and/or by increasing airflow.

Table 10. 132-Lead PGA Package Thermal Characteristics

Thermal Resistance — °C/Watt						
Parameter	**Airflow — ft./min (m/sec)**					
	0 (0)	**200 (1.01)**	**400 (2.03)**	**600 (3.04)**	**800 (4.06)**	**1000 (5.08)**
θ_{JC} (Junction-to-Case)	3	3	3	3	3	3
θ_{CA} (Case-to-Ambient) (No Heatsink)	18	15	12	11	11	11
θ_{CA} (Case-to-Ambient) (Omnidirectional Heatsink)	15	12	9	8	8	8
θ_{CA} (Case-to-Ambient) (Unidirectional Heatsink)	14	11	8	7	7	7

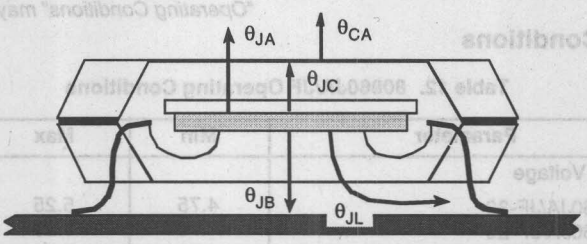


NOTES:

1. This table applies to a PGA device plugged into a socket or soldered directly into a board.
2. $\theta_{JA} = \theta_{JC} + \theta_{CA}$
3. $\theta_{J\text{-}CAP}$ = 4°C/W (approx.)
4. $\theta_{J\text{-}PIN}$ = 4°C/W (inner pins) (approx.)
5. $\theta_{J\text{-}PIN}$ = 8°C/W (outer pins) (approx.)

Table 11. 132-Lead PQFP Package Thermal Characteristics

Thermal Resistance — °C/Watt							
	Airflow — ft./min (m/sec)						
Parameter	0 (0)	50 (0.25)	100 (0.50)	200 (1.01)	400 (2.03)	600 (3.04)	800 (4.06)
θ_{JC} (Junction-to-Case)	6	7	7	7	7	7	7
θ_{CA} (Case-to-Ambient; No Heatsink)	23	20	18	14	10	9	8



NOTES:

1. This table applies to a PQFP device soldered directly into board.
2. $\theta_{JA} = \theta_{JC} + \theta_{CA}$
3. θ_{JL} = 18°C/W (approx.)
4. θ_{JB} = 18°C/W (approx.)

4.0 ELECTRICAL SPECIFICATIONS

4.1 Absolute Maximum Ratings

Parameter	Maximum Rating
Storage Temperature	–65° C to +150° C
Case Temperature Under Bias	–65° C to +110° C
Supply Voltage wrt. V_{SS}	–0.5V to + 4.6V
Voltage on Other Pins wrt. V_{SS}	–0.5V to V_{CC} + 0.5V

NOTICE: This data sheet contains preliminary information on new products in production. The specifications are subject to change without notice.

WARNING: *Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

4.2 Operating Conditions

Table 12. 80960JA/JF Operating Conditions

Symbol	Parameter	Min	Max	Units	Notes
V_{CC}	Supply Voltage				
	80960JA/JF-33	4.75	5.25	V	
	80960JA/JF-25	4.75	5.25		
	80960JA/JF-16	4.75	5.25		
f_{CLKIN}	Input Clock Frequency				
	80960JA/JF-33	8	33.33	MHz	
	80960JA/JF-25	8	25		
	80960JA/JF-16	8	16.67		
T_C	Operating Case Temperature				
	A80960JA/JF-33 (132 PGA)	0	100	°C	
	A80960JA/JF-25 (132 PGA)	0	100		
	A80960JA/JF-16 (132 PGA)	0	100		
	NG80960JA/JF-33 (132 PQFP)	0	100		
	NG80960JA/JF-25 (132 PQFP)	0	100		
	NG80960JA/JF-16 (132 PQFP)	0	100		

4.3 Connection Recommendations

For clean on-chip power distribution, V_{CC} and V_{SS} pins separately feed the device's functional units. Power and ground connections must be made to all 80960JA/JF power and ground pins. On the circuit board, every V_{CC} pin should connect to a power plane and every V_{SS} pin should connect to a ground plane. Place liberal decoupling capacitance near the 80960JA/JF, since the processor can cause transient power surges.

Pay special attention to the Test Reset ($\overline{TRST}$) pin. It is essential that the JTAG Boundary Scan Test Access Port (TAP) controller initializes to a known state whether it will be used or not. If the JTAG Boundary Scan function will be used, connect a pulldown resistor between the $\overline{TRST}$ pin and V_{SS}. If the JTAG Boundary Scan function will not be used (even for board-level testing), connect the $\overline{TRST}$ pin to V_{SS}. Also, do not connect the TDI, TDO, and TCK pins if the TAP Controller will not be used.

Pins identified as NC must not be connected in the system.

4.4 DC Specifications

Table 13. 80960JA/JF DC Characteristics

Symbol	Parameter	Min	Typ	Max	Units	Notes
V_{IL}	Input Low Voltage	-0.3		0.8	V	
V_{IH}	Input High Voltage	2.0		V_{CC} + 0.3	V	
V_{OL}	Output Low Voltage			0.45	V	I_{OL} = 5 mA
V_{OH}	Output High Voltage	2.4 V_{CC} - 0.5			V	I_{OH} = -1 mA I_{OH} = -200 µA
V_{OLP}	Output Ground Bounce		< 0.8		V	(1,2)
C_{IN}	Input Capacitance PGA PQFP			 12 10	pF	$f_{CLKIN} = f_{MIN}$ (2)
C_{OUT}	I/O or Output Capacitance PGA PQFP			 12 10	pF	$f_{CLKIN} = f_{MIN}$ (2)
C_{CLK}	CLKIN Capacitance PGA PQFP			 12 10	pF	$f_{CLKIN} = f_{MIN}$ (2)

NOTES:
1. Typical is measured with V_{CC} = 5.0V and temperature = 25 °C.
2. Not tested.

Table 14. 80960JA/JF I_{CC} Characteristics (Sheet 1 of 2)

Symbol	Parameter	Typ	Max	Units	Notes
I_{LI1}	Input Leakage Current for each pin except TCK, TDI, $\overline{TRST}$ and TMS		± 1	µA	$0 \le V_{IN} \le V_{CC}$
I_{LI2}	Input Leakage Current for TCK, TDI, $\overline{TRST}$ and TMS	-140	-250	µA	V_{IN} = 0.45V (1)
I_{LO}	Output Leakage Current		± 1	µA	$0.4 \le V_{OUT} \le V_{CC}$
I_{CC} Active (Power Supply)	80960JA/JF-33 80960JA/JF-25 80960JA/JF-16		515 390 245	mA	(2,3) (2,3) (2,3)
I_{CC} Active (Thermal)	80960JA/JF-33 80960JA/JF-25 80960JA/JF-16	420 325 205		mA	(2,4) (2,4) (2,4)

Table 14. 80960JA/JF I_{CC} Characteristics (Sheet 2 of 2)

Symbol	Parameter	Typ	Max	Units	Notes
I_{CC} Test (Power modes)	Reset mode				
	80960JA/JF-33		380	mA	(5)
	80960JA/JF-25		300		(5)
	80960JA/JF-16		210		(5)
	Halt mode				
	80960JA/JF-33		40		(5)
	80960JA/JF-25		30		(5)
	80960JA/JF-16		20		(5)
	ONCE mode		10		(5)

NOTES:

1. These pins have internal pullup devices. Typical leakage current is not tested.
2. Measured with device operating and outputs loaded to the test condition in Figure 6, AC Test Load (pg. 32).
3. I_{CC} Active (Power Supply) value is provided for selecting your system's power supply. It is measured using one of the worst case instruction mixes with V_{CC} = 5.3V. This parameter is characterized but not tested.
4. I_{CC} Active (Thermal) value is provided for your system's thermal management. Typical I_{CC} is measured with V_{CC} = 5.0V and temperature = 25° C. This parameter is characterized but not tested.
5. I_{CC} Test (Power modes) refers to the I_{CC} values that are tested when the 80960JA/JF is in Reset mode, Halt mode or ONCE mode with V_{CC} = 5.3V.

T_C	CLKIN Period	30	62.5	ns	
T_{CS}	CLKIN Period Stability		± 250	ps	(1, 2)
T_{CH}	CLKIN High Time	12		ns	Measured at 1.5 V (1)
T_{CL}	CLKIN Low Time	12		ns	Measured at 1.5 V (1)
T_{CR}	CLKIN Rise Time		4	ns	0.8 V to 2.0 V (1)
T_{CF}	CLKIN Fall Time		4	ns	2.0 V to 0.8 V (1)
	SYNCHRONOUS OUTPUT TIMINGS				
T_{OV1}	Output Valid Delay, Except ALE/$\overline{ALE}$ Inactive and DT/$\overline{R}$	3.5	15	ns	(3)
T_{OV2}	Output Valid Delay, DT/$\overline{R}$	$0.5T_C + 3.5$	$0.5T_C + 15$	ns	
T_{OF}	Output Float Delay	3.5	13	ns	(4)
	SYNCHRONOUS INPUT TIMINGS				
T_{IS1}	Input Setup to CLKIN — AD31:0, $\overline{NMI}$, $\overline{XINT7:0}$	6		ns	(5)
T_{IH1}	Input Hold from CLKIN — AD31:0, $\overline{NMI}$, $\overline{XINT7:0}$	2		ns	(5)
T_{IS2}	Input Setup to CLKIN — $\overline{RDYRCV}$ and HOLD	7.5		ns	(6)
T_{IH2}	Input Hold from CLKIN — $\overline{RDYRCV}$ and HOLD	1		ns	(6)
T_{IS3}	Input Setup to CLKIN — $\overline{RESET}$	6		ns	(7)
T_{IH3}	Input Hold from CLKIN — $\overline{RESET}$	2		ns	(7)
T_{IS4}	Input Setup to $\overline{RESET}$ — $\overline{ONCE}$, STEST	6		ns	(8)
T_{IH4}	Input Hold from $\overline{RESET}$ — $\overline{ONCE}$, STEST	2		ns	(8)

NOTE: See Table 16 on page 27 for note definitions for this table.

Table 15. 80960JA/JF AC Characteristics (33 MHz) (Sheet 2 of 2)

Symbol	Parameter	Min	Max	Units	Notes
RELATIVE OUTPUT TIMINGS					
T_{LXL}	ALE/$\overline{ALE}$ Width	$0.5T_C - 7$		ns	(9)
T_{LXA}	Address Hold from ALE/$\overline{ALE}$ Inactive				Equal Loading (9)
T_{DXD}	DT/$\overline{R}$ Valid to $\overline{DEN}$ Active				Equal Loading (9)
BOUNDARY SCAN TEST SIGNAL TIMINGS					
T_{BSF}	TCK Frequency		$0.5T_F$	MHz	
T_{BSCH}	TCK High Time	15		ns	Measured at 1.5 V (1)
T_{BSCL}	TCK Low Time	15		ns	Measured at 1.5 V (1)
T_{BSCR}	TCK Rise Time		5	ns	0.8 V to 2.0 V (1)
T_{BSCF}	TCK Fall Time		5	ns	2.0 V to 0.8 V (1)
T_{BSIS1}	Input Setup to TCK — TDI, TMS	4		ns	
T_{BSIH1}	Input Hold from TCK — TDI, TMS	6		ns	
T_{BSOV1}	TDO Valid Delay	3	30	ns	(1, 10)
T_{BSOF1}	TDO Float Delay	3	30	ns	(1, 10)
T_{BSOV2}	All Outputs (Non-Test) Valid Delay	3	30	ns	(1, 10)
T_{BSOF2}	All Outputs (Non-Test) Float Delay	3	30	ns	(1, 10)
T_{BSIS2}	Input Setup to TCK — All Inputs (Non-Test)	4		ns	
T_{BSIH2}	Input Hold from TCK — All Inputs (Non-Test)	6		ns	

NOTE: See Table 16 on page 27 for note definitions for this table.

Table 16. Note Definitions for Table 15, 80960JA/JF AC Characteristics (33 MHz) (pg. 25)

NOTES:

1. Not tested.
2. To ensure a 1:1 relationship between the amplitude of the input jitter and the internal clock, the jitter frequency spectrum should not have any power peaking between 500 KHz and 1/3 of the CLKIN frequency.
3. Inactive ALE/$\overline{ALE}$ refers to the falling edge of ALE and the rising edge of $\overline{ALE}$. For inactive ALE/$\overline{ALE}$ timings, refer to Relative Output Timings in this table.
4. A float condition occurs when the output current becomes less than I_{LO}. Float delay is not tested, but is designed to be no longer than the valid delay.
5. AD31:0 are synchronous inputs. Setup and hold times must be met for proper processor operation. $\overline{NMI}$ and XINT7:0 may be synchronous or asynchronous. Meeting setup and hold time guarantees recognition at a particular clock edge. For asynchronous operation, NMI and XINT7:0 must be asserted for a minimum of two CLKIN periods to guarantee recognition.
6. $\overline{RDYRCV}$ and HOLD are synchronous inputs. Setup and hold times must be met for proper processor operation.
7. $\overline{RESET}$ may be synchronous or asynchronous. Meeting setup and hold time guarantees recognition at a particular clock edge.
8. $\overline{ONCE}$ and $\overline{STEST}$ must be stable at the rising edge of $\overline{RESET}$ for proper operation.
9. Guaranteed by design. May not be 100% tested.
10. Relative to falling edge of TCK.

Table 17. 80960JA/JF AC Characteristics (25 MHz) (Sheet 1 of 3)

Symbol	Parameter	Min	Max	Units	Notes
	INPUT CLOCK TIMINGS				
T_F	CLKIN Frequency	8	25	MHz	
T_C	CLKIN Period	40	62.5	ns	
T_{CS}	CLKIN Period Stability		± 250	ps	(1, 2)
T_{CH}	CLKIN High Time	16		ns	Measured at 1.5 V (1)
T_{CL}	CLKIN Low Time	16		ns	Measured at 1.5 V (1)
T_{CR}	CLKIN Rise Time		5	ns	0.8 V to 2.0 V (1)
T_{CF}	CLKIN Fall Time		5	ns	2.0 V to 0.8 V (1)
	SYNCHRONOUS OUTPUT TIMINGS				
T_{OV1}	Output Valid Delay, Except ALE/$\overline{ALE}$ Inactive and DT/$\overline{R}$	3.5	17	ns	(3)
T_{OV2}	Output Valid Delay, DT/$\overline{R}$	$0.5T_C + 3.5$	$0.5T_C + 17$	ns	
T_{OF}	Output Float Delay	3.5	15	ns	(4)

Table 17. 80960JA/JF AC Characteristics (25 MHz) (Sheet 2 of 3)

Symbol	Parameter	Min	Max	Units	Notes
	SYNCHRONOUS INPUT TIMINGS				
T_{IS1}	Input Setup to CLKIN — AD31:0, $\overline{NMI}$, $\overline{XINT7:0}$	8		ns	(5)
T_{IH1}	Input Hold from CLKIN — AD31:0, $\overline{NMI}$, $\overline{XINT7:0}$	2		ns	(5)
T_{IS2}	Input Setup to CLKIN — $\overline{RDYRCV}$ and HOLD	8		ns	(6)
T_{IH2}	Input Hold from CLKIN — $\overline{RDYRCV}$ and HOLD	1		ns	(6)
T_{IS3}	Input Setup to CLKIN — $\overline{RESET}$	8		ns	(7)
T_{IH3}	Input Hold from CLKIN — $\overline{RESET}$	2		ns	(7)
T_{IS4}	Input Setup to $\overline{RESET}$ — $\overline{ONCE}$, STEST	8		ns	(8)
T_{IH4}	Input Hold from $\overline{RESET}$ — $\overline{ONCE}$, STEST	2		ns	(8)
	RELATIVE OUTPUT TIMINGS				
T_{LXL}	ALE/$\overline{ALE}$ Width	$0.5T_C$ - 7.5		ns	(9)
T_{LXA}	Address Hold from ALE/$\overline{ALE}$ Inactive				Equal Loading (9)
T_{DXD}	DT/$\overline{R}$ Valid to $\overline{DEN}$ Active				Equal Loading (9)
	BOUNDARY SCAN TEST SIGNAL TIMINGS				
T_{BSF}	TCK Frequency		$0.5T_F$	MHz	
T_{BSCH}	TCK High Time	15		ns	Measured at 1.5 V (1)
T_{BSCL}	TCK Low Time	15		ns	Measured at 1.5 V (1)
T_{BSCR}	TCK Rise Time		5	ns	0.8 V to 2.0 V (1)
T_{BSCF}	TCK Fall Time		5	ns	2.0 V to 0.8 V (1)
T_{BSIS1}	Input Setup to TCK — TDI, TMS	4		ns	
T_{BSIH1}	Input Hold from TCK — TDI, TMS	6		ns	
T_{BSOV1}	TDO Valid Delay	3	30	ns	(1, 10)
T_{BSOF1}	TDO Float Delay	3	30	ns	(1, 10)

Table 17. 80960JA/JF AC Characteristics (25 MHz) (Sheet 3 of 3)

Symbol	Parameter	Min	Max	Units	Notes
T_{BSOV2}	All Outputs (Non-Test) Valid Delay	3	30	ns	(1, 10)
T_{BSOF2}	All Outputs (Non-Test) Float Delay	3	30	ns	(1, 10)
T_{BSIS2}	Input Setup to TCK — All Inputs (Non-Test)	4		ns	
T_{BSIH2}	Input Hold from TCK — All Inputs (Non-Test)	6		ns	

NOTES:

1. Not tested.
2. To ensure a 1:1 relationship between the amplitude of the input jitter and the internal clock, the jitter frequency spectrum should not have any power peaking between 500 KHz and 1/3 of the CLKIN frequency.
3. Inactive ALE/$\overline{\text{ALE}}$ refers to the falling edge of ALE and the rising edge of $\overline{\text{ALE}}$. For inactive ALE/$\overline{\text{ALE}}$ timings, refer to Relative Output Timings in this table.
4. A float condition occurs when the output current becomes less than I_{LO}. Float delay is not tested, but is designed to be no greater than the valid delay.
5. AD31:0 are synchronous inputs. Setup and hold times must be met for proper processor operation. $\overline{\text{NMI}}$ and XINT7:0 may be synchronous or asynchronous. Meeting setup and hold time guarantees recognition at a particular clock edge. For asynchronous operation, $\overline{\text{NMI}}$ and XINT7:0 must be asserted for a minimum of two CLKIN periods to guarantee recognition.
6. RDYRCV and HOLD are synchronous inputs. Setup and hold times must be met for proper processor operation.
7. $\overline{\text{RESET}}$ may be synchronous or asynchronous. Meeting setup and hold time guarantees recognition at a particular clock edge.
8. ONCE and $\overline{\text{STEST}}$ must be stable at the rising edge of $\overline{\text{RESET}}$ for proper operation.
9. Guaranteed by design. May not be 100% tested.
10. Relative to falling edge of TCK.

Table 18. 80960JA/JF AC Characteristics (16 MHz) (Sheet 1 of 3)

Symbol	Parameter	Min	Max	Units	Notes
	INPUT CLOCK TIMINGS				
T_F	CLKIN Frequency	8	16.67	MHz	
T_C	CLKIN Period	60	62.5	ns	
T_{CS}	CLKIN Period Stability		±250	ps	(1, 2)
T_{CH}	CLKIN High Time	24		ns	Measured at 1.5 V (1)
T_{CL}	CLKIN Low Time	24		ns	Measured at 1.5 V (1)
T_{CR}	CLKIN Rise Time		8	ns	0.8 V to 2.0 V (1)
T_{CF}	CLKIN Fall Time		8	ns	2.0 V to 0.8 V (1)

Table 18. 80960JA/JF AC Characteristics (16 MHz) (Sheet 2 of 3)

Symbol	Parameter	Min	Max	Units	Notes
	SYNCHRONOUS OUTPUT TIMINGS				
T_{OV1}	Output Valid Delay, Except ALE/$\overline{ALE}$ Inactive and DT/$\overline{R}$	3.5	18	ns	(3)
T_{OV2}	Output Valid Delay, DT/$\overline{R}$	$0.5T_C + 3.5$	$0.5T_C + 18$	ns	
T_{OF}	Output Float Delay	3.5	16	ns	(4)
	SYNCHRONOUS INPUT TIMINGS				
T_{IS1}	Input Setup to CLKIN — AD31:0, $\overline{NMI}$, $\overline{XINT7:0}$	8		ns	(5)
T_{IH1}	Input Hold from CLKIN — AD31:0, $\overline{NMI}$, $\overline{XINT7:0}$	2		ns	(5)
T_{IS2}	Input Setup to CLKIN — $\overline{RDYRCV}$ and HOLD	8		ns	(6)
T_{IH2}	Input Hold from CLKIN — $\overline{RDYRCV}$ and HOLD	1		ns	(6)
T_{IS3}	Input Setup to CLKIN — $\overline{RESET}$	8		ns	(7)
T_{IH3}	Input Hold from CLKIN — $\overline{RESET}$	2		ns	(7)
T_{IS4}	Input Setup to $\overline{RESET}$ — $\overline{ONCE}$, STEST	8		ns	(8)
T_{IH4}	Input Hold from $\overline{RESET}$ — $\overline{ONCE}$, STEST	2		ns	(8)
	RELATIVE OUTPUT TIMINGS				
T_{LXL}	ALE/$\overline{ALE}$ Width	$0.5T_C - 8$		ns	(9)
T_{LXA}	Address Hold from ALE/$\overline{ALE}$ Inactive				Equal Loading (9)
T_{DXD}	DT/$\overline{R}$ Valid to $\overline{DEN}$ Active				Equal Loading (9)
	BOUNDARY SCAN TEST SIGNAL TIMINGS				
T_{BSF}	TCK Frequency		$0.5T_F$	MHz	
T_{BSCH}	TCK High Time	15		ns	Measured at 1.5 V (1)
T_{BSCL}	TCK Low Time	15		ns	Measured at 1.5 V (1)
T_{BSCR}	TCK Rise Time		5	ns	0.8 V to 2.0 V (1)
T_{BSCF}	TCK Fall Time		5	ns	2.0 V to 0.8 V (1)
T_{BSIS1}	Input Setup to TCK — TDI, TMS	4		ns	

T_{BSOV2}	All Outputs (Non-Test) Valid Delay	3	30	ns	(1, 10)
T_{BSOF2}	All Outputs (Non-Test) Float Delay	3	36	ns	(1, 10)
T_{BSIS2}	Input Setup to TCK — All Inputs (Non-Test)	4		ns	
T_{BSIH2}	Input Hold from TCK — All Inputs (Non-Test)	6		ns	

NOTES:

1. Not tested.
2. To ensure a 1:1 relationship between the amplitude of the input jitter and the internal clock, the jitter frequency spectrum should not have any power peaking between 500 KHz and 1/3 of the CLKIN frequency.
3. Inactive ALE/$\overline{\text{ALE}}$ refers to the falling edge of ALE and the rising edge of $\overline{\text{ALE}}$. For inactive ALE/$\overline{\text{ALE}}$ timings, refer to Relative Output Timings in this table.
4. A float condition occurs when the output current becomes less than I_{LO}. Float delay is not tested, but is designed to be no greater than the valid delay.
5. AD31:0 are synchronous inputs. Setup and hold times must be met for proper processor operation. $\overline{\text{NMI}}$ and $\overline{\text{XINT}}$7:0 may be synchronous or asynchronous. Meeting setup and hold time guarantees recognition at a particular clock edge. For asynchronous operation, $\overline{\text{NMI}}$ and $\overline{\text{XINT}}$7:0 must be asserted for a minimum of two CLKIN periods to guarantee recognition.
6. $\overline{\text{RDYRCV}}$ and HOLD are synchronous inputs. Setup and hold times must be met for proper processor operation.
7. $\overline{\text{RESET}}$ may be synchronous or asynchronous. Meeting setup and hold time guarantees recognition at a particular clock edge.
8. $\overline{\text{ONCE}}$ and $\overline{\text{STEST}}$ must be stable at the rising edge of $\overline{\text{RESET}}$ for proper operation.
9. Guaranteed by design. May not be 100% tested.
10. Relative to falling edge of TCK.

4.5.1 AC Test Conditions and Derating Curves

The AC Specifications in **Section 4.5, AC Specifications** are tested with the 50 pF load indicated in Figure 6. Figure 7 shows how timings vary with load capacitance; Figure 8 shows how output rise and fall times vary with load capacitance.

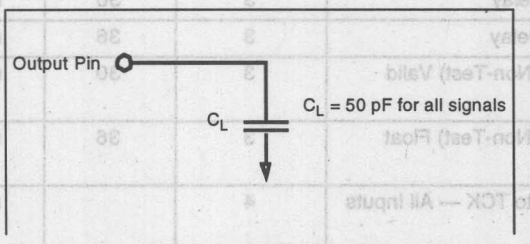


Figure 6. AC Test Load

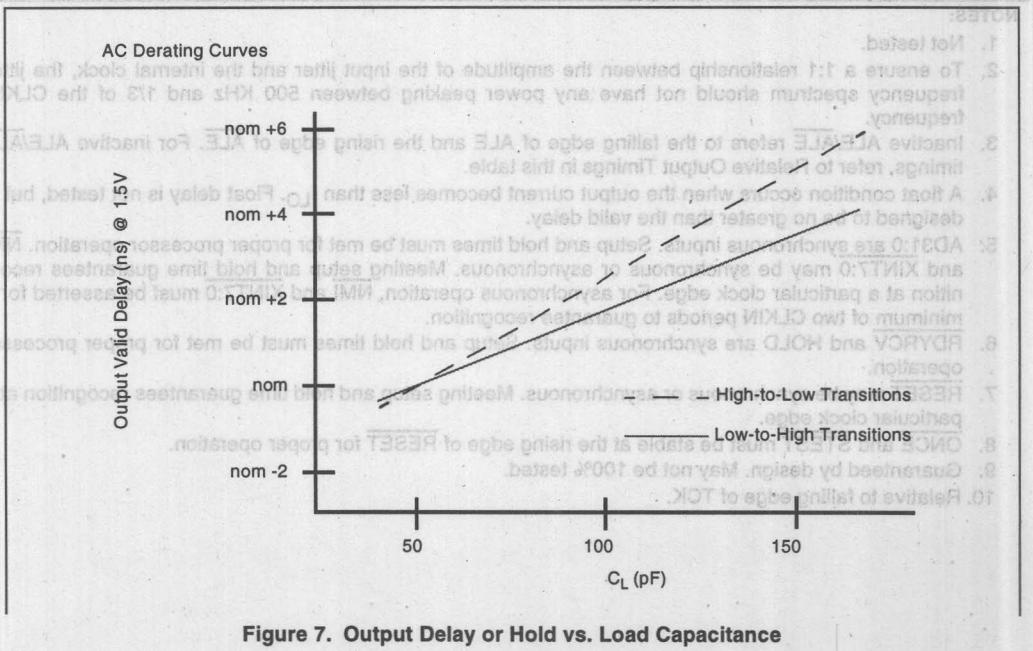


Figure 7. Output Delay or Hold vs. Load Capacitance

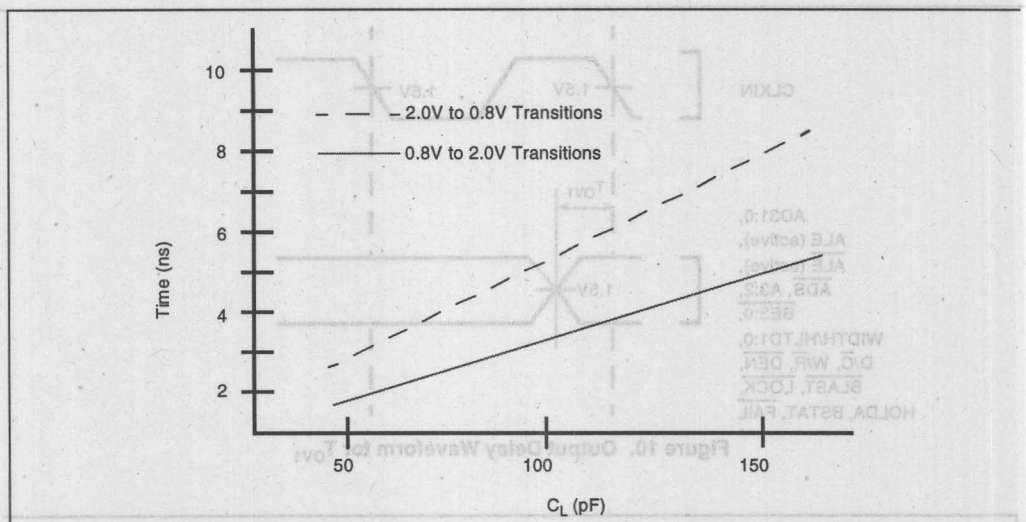


Figure 8. Rise and Fall Time Derating

4.5.2 AC Timing Waveforms

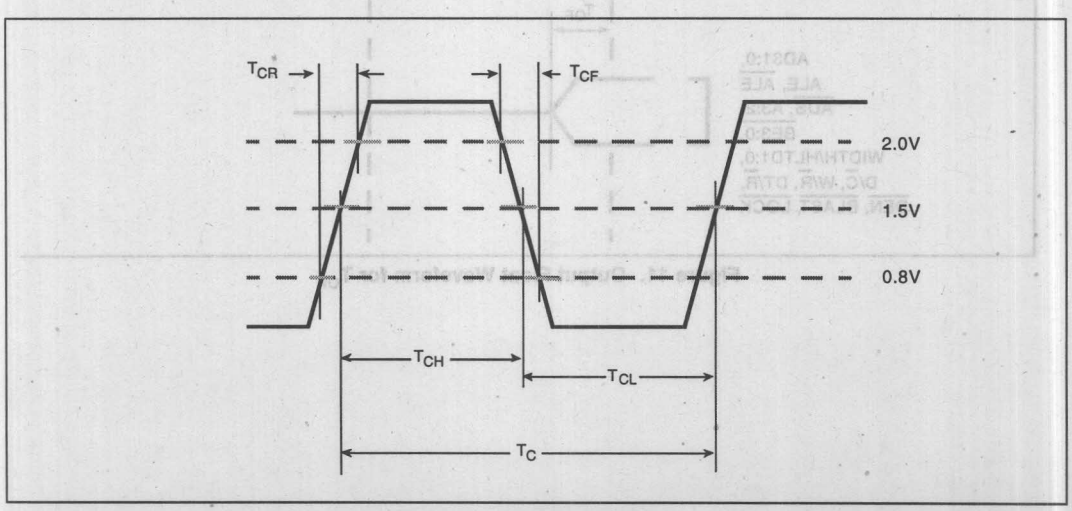


Figure 9. CLKIN Waveform

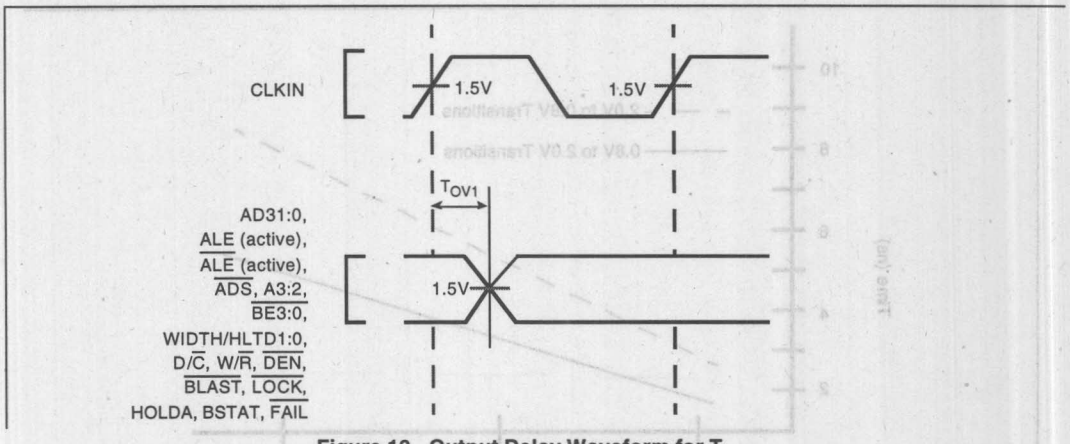


Figure 10. Output Delay Waveform for T_{OV1}

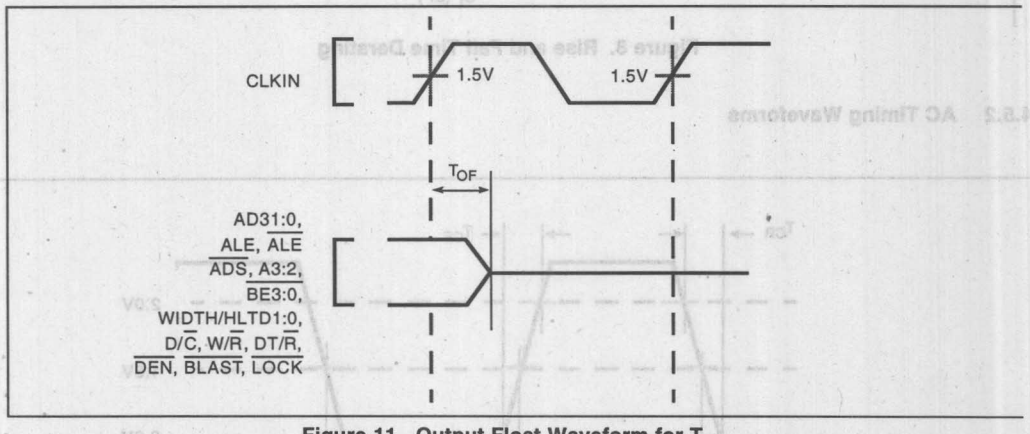


Figure 11. Output Float Waveform for T_{OF}

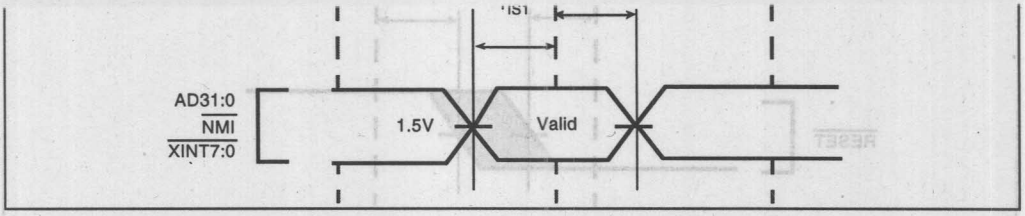


Figure 12. Input Setup and Hold Waveform for T_{IS1} and T_{IH1}

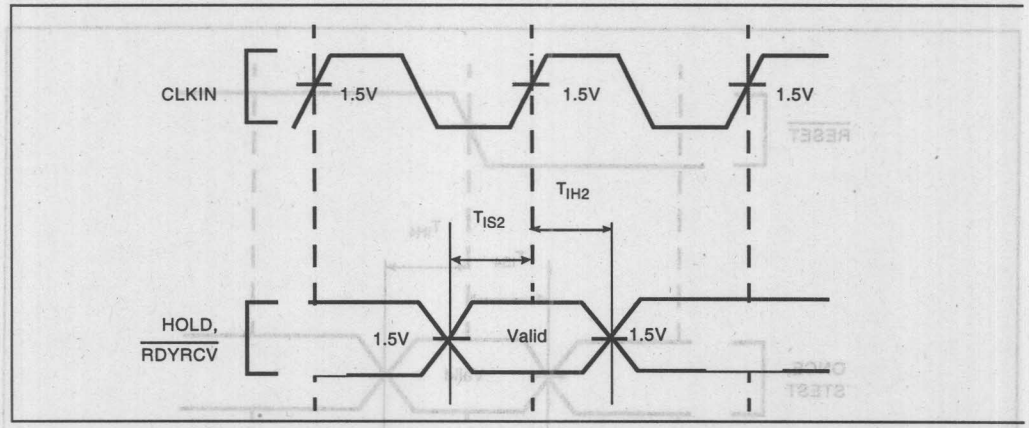


Figure 13. Input Setup and Hold Waveform for T_{IS2} and T_{IH2}

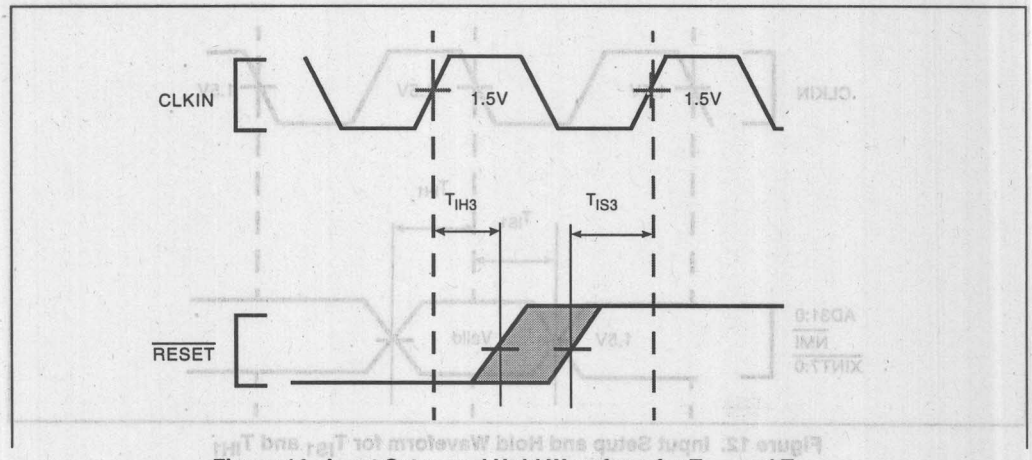


Figure 14. Input Setup and Hold Waveform for T_{IS3} and T_{IH3}

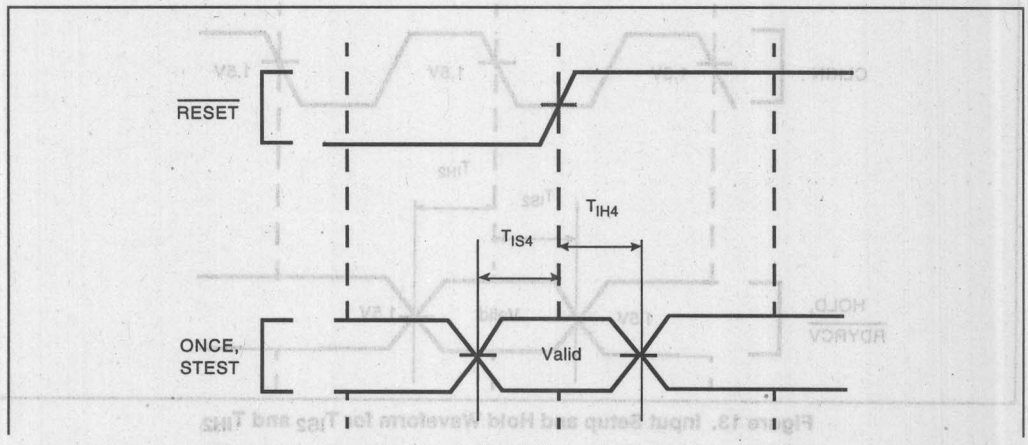


Figure 15. Input Setup and Hold Waveform for T_{IS4} and T_{IH4}

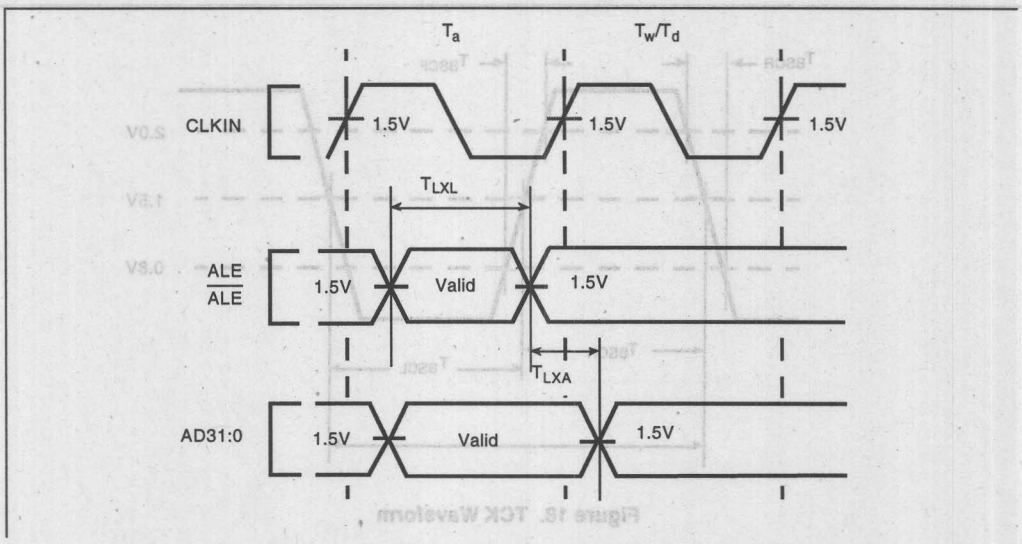


Figure 16. Relative Timings Waveform for T_{LXL} and T_{LXA}

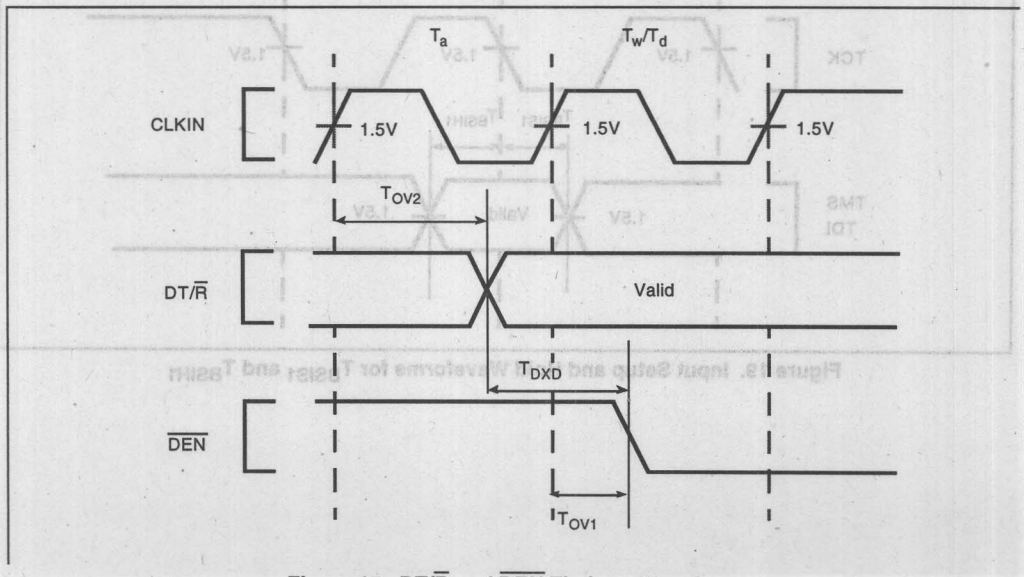


Figure 17. DT/$\overline{R}$ and $\overline{DEN}$ Timings Waveform

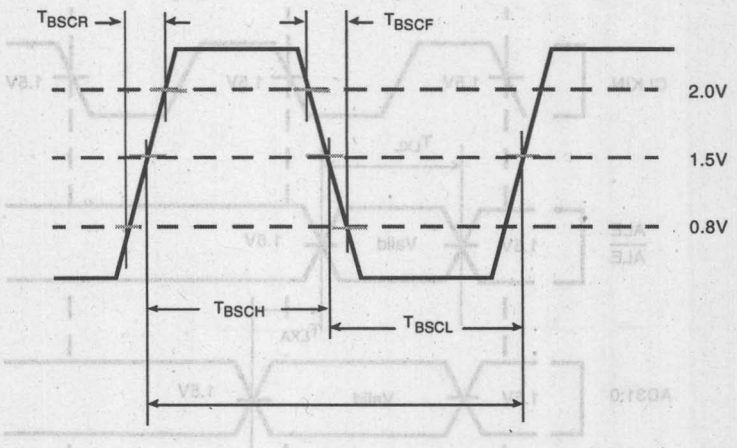


Figure 18. TCK Waveform

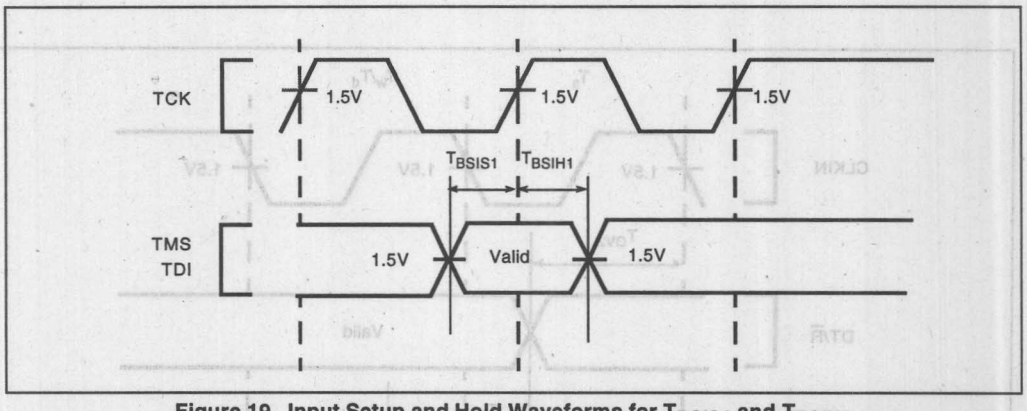


Figure 19. Input Setup and Hold Waveforms for T_{BSIS1} and T_{BSIH1}

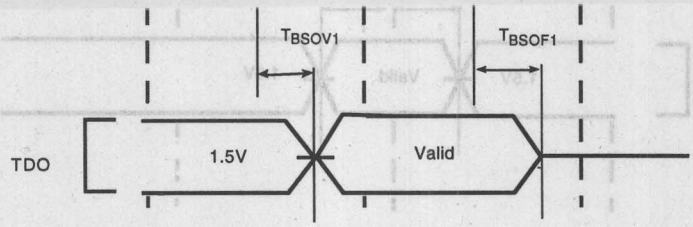


Figure 20. Output Delay and Output Float Waveform for T_{BSOV1} AND T_{BSOF1}

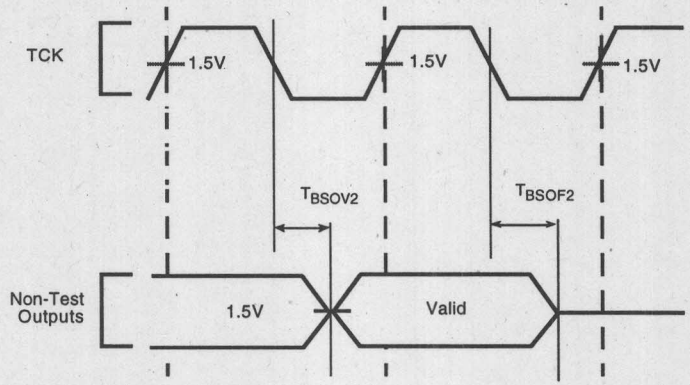


Figure 21. Output Delay and Output Float Waveform for T_{BSOV2} and T_{BSOF2}

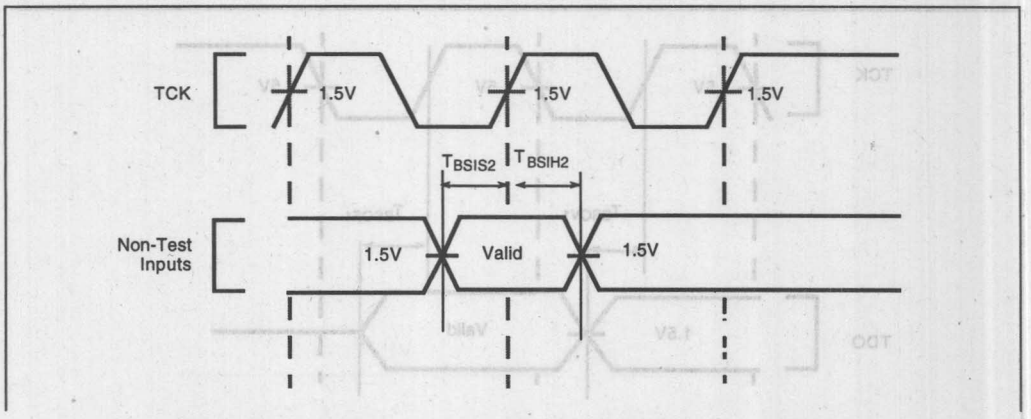


Figure 22. Input Setup and Hold Waveform for T_{BSIS2} and T_{BSIH2}

5.0 BUS FUNCTIONAL WAVEFORMS

Figures 23 through 28 illustrate typical 80960JA/JF bus transactions. Figure 29 depicts the bus arbitration sequence. Figure 30 illustrates the processor reset sequence from the time power is applied to the device. Figure 31 illustrates the processor reset sequence when the processor is in operation. Figure 32 illustrates the processor ONCE sequence from the time power is applied to the device. Figures 33 and 34 also show accesses on 32-bit buses. Tables 19 through 22 summarize all possible combinations of bus accesses across 8-, 16-, and 32-bit buses according to data alignment.

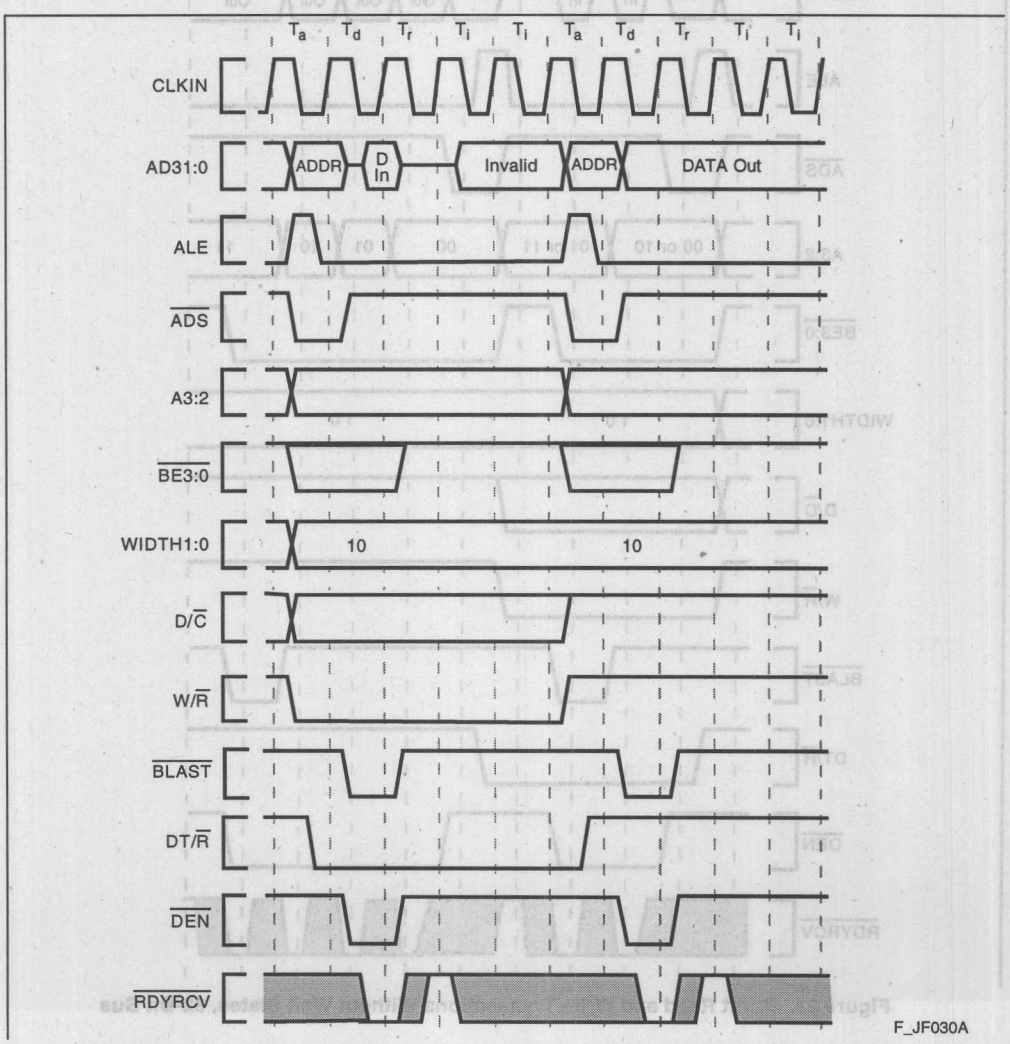


Figure 23. Non-Burst Read and Write Transactions Without Wait States, 32-Bit Bus

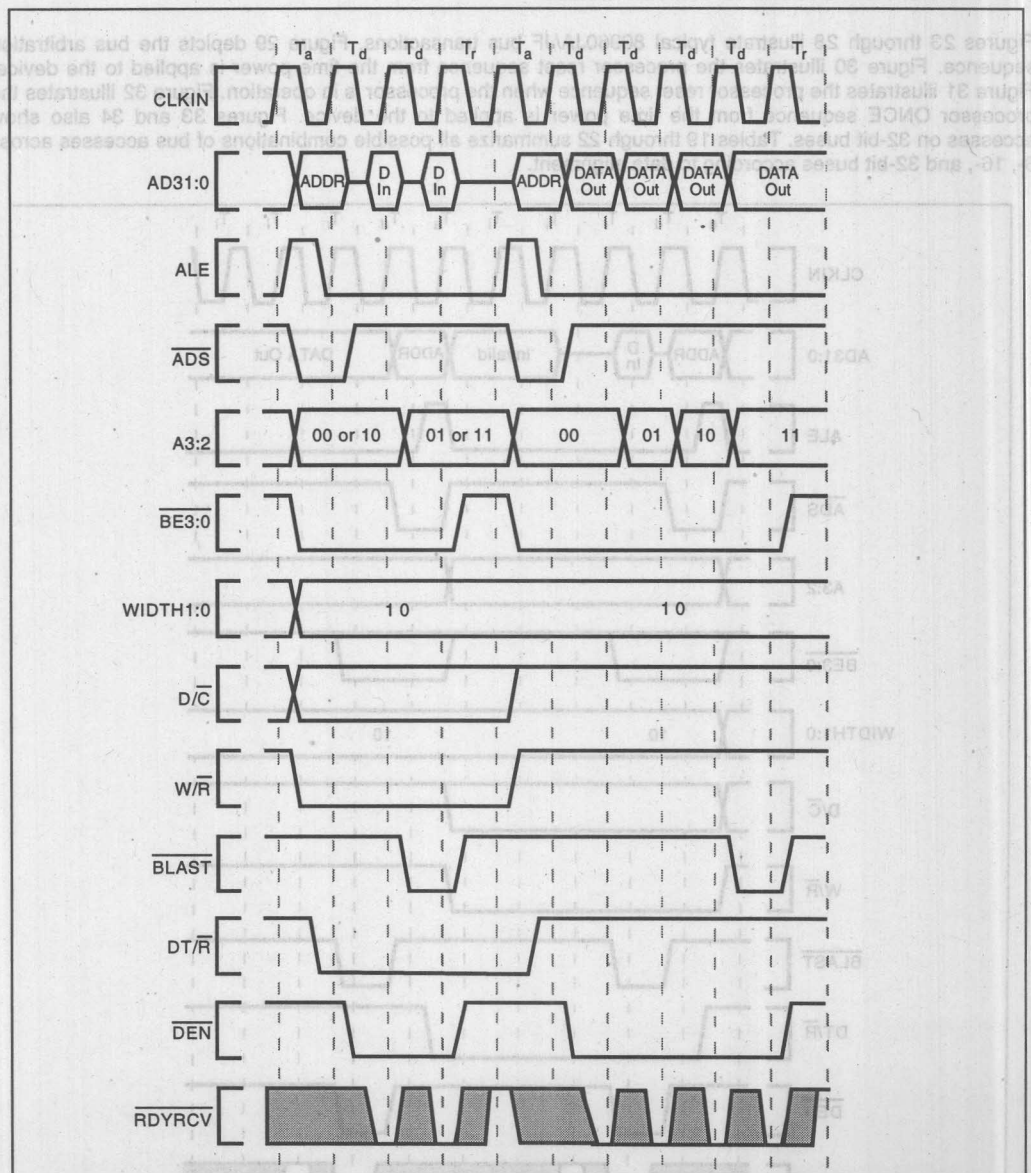


Figure 24. Burst Read and Write Transactions Without Wait States, 32-Bit Bus

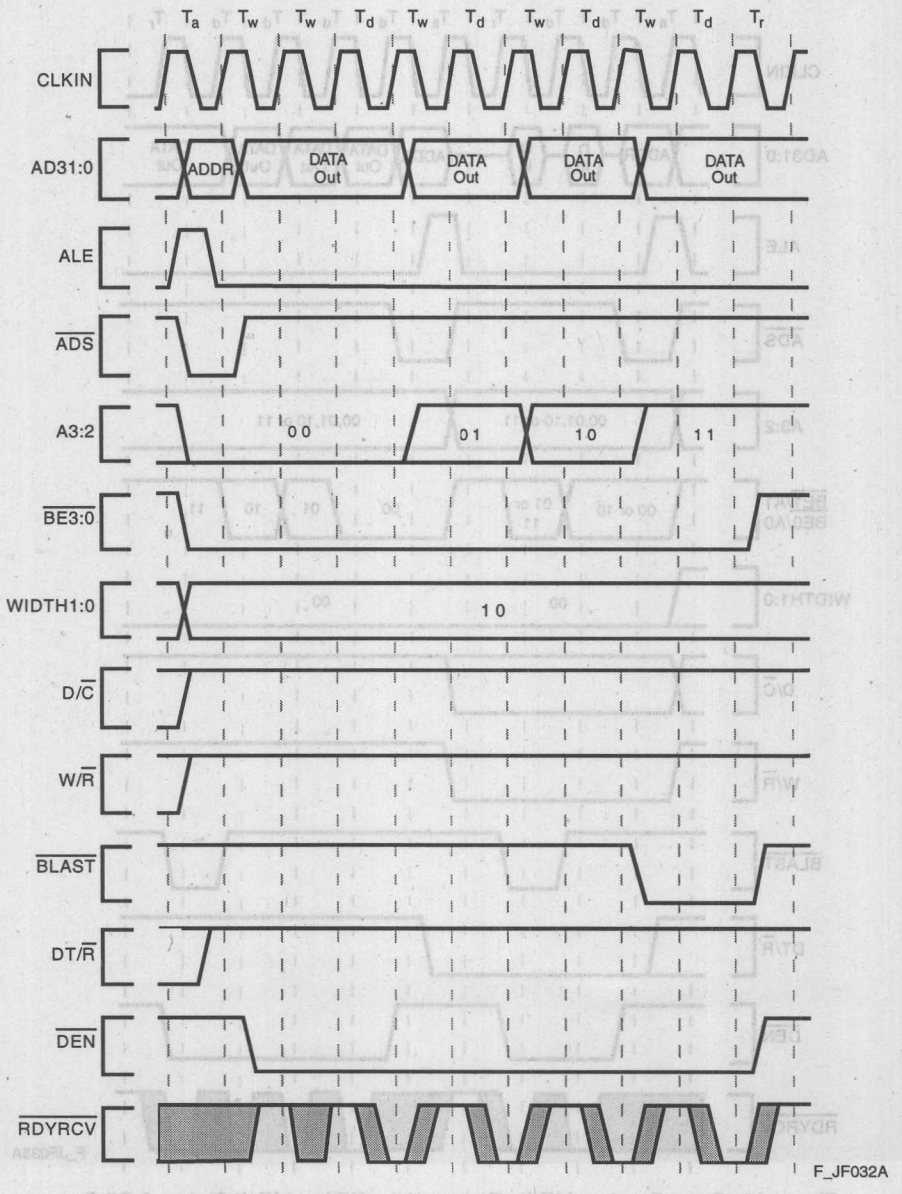


Figure 25. Burst Write Transactions With 2,1,1,1 Wait States, 32-Bit Bus

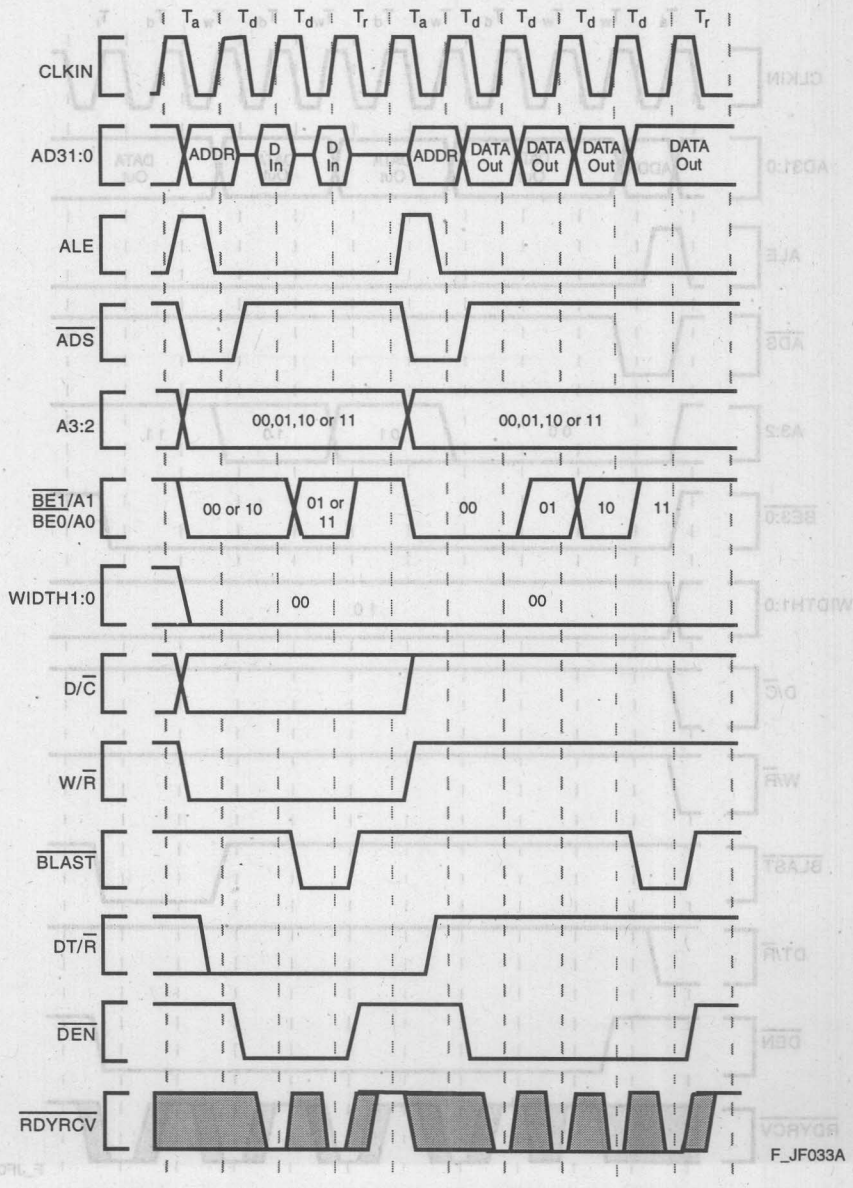


Figure 26. Burst Read and Write Transactions Without Wait States, 8-Bit Bus

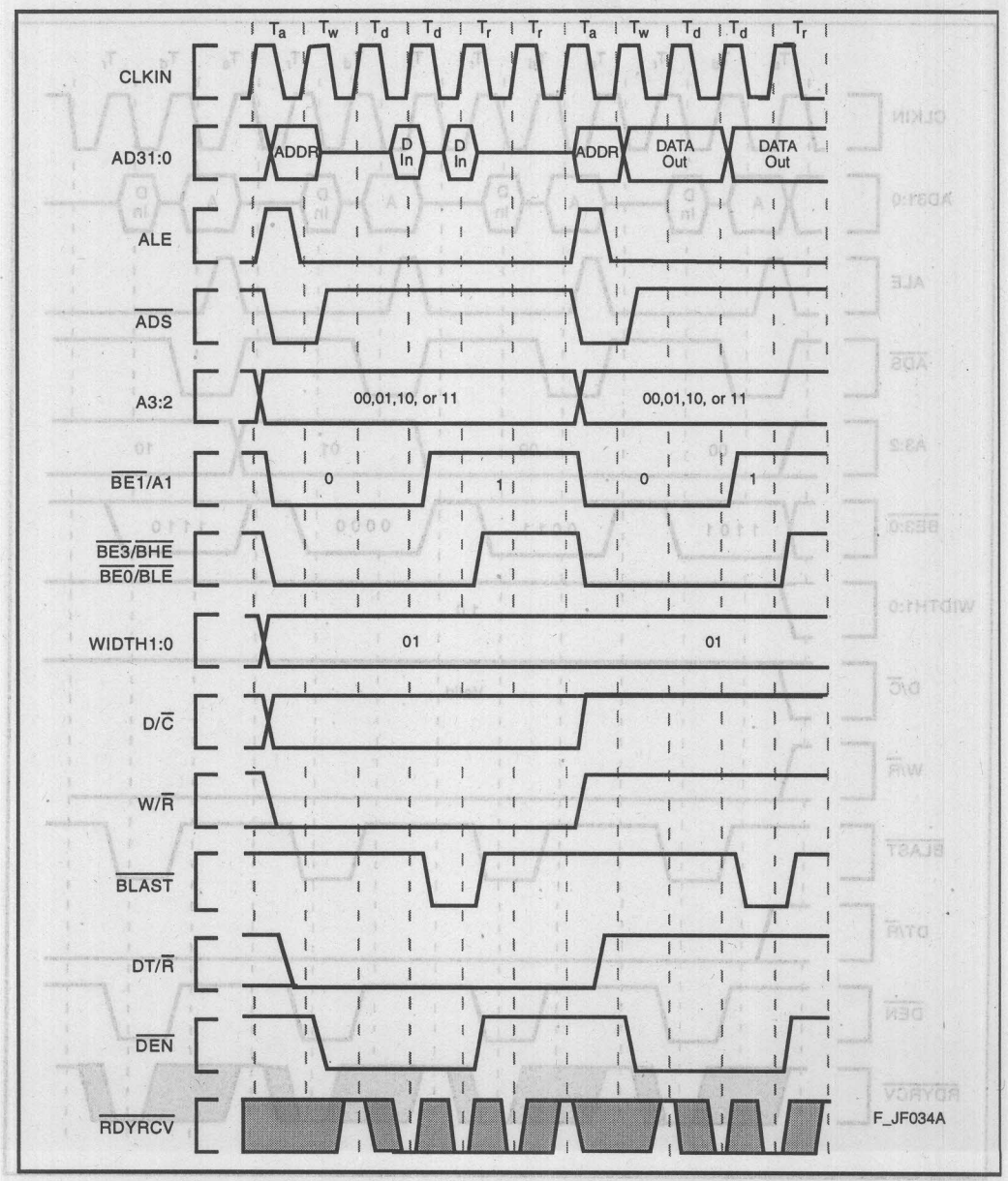


Figure 27. Burst Read and Write Transactions With 1, 0 Wait States and Extra Tr State on Read, 16-Bit Bus

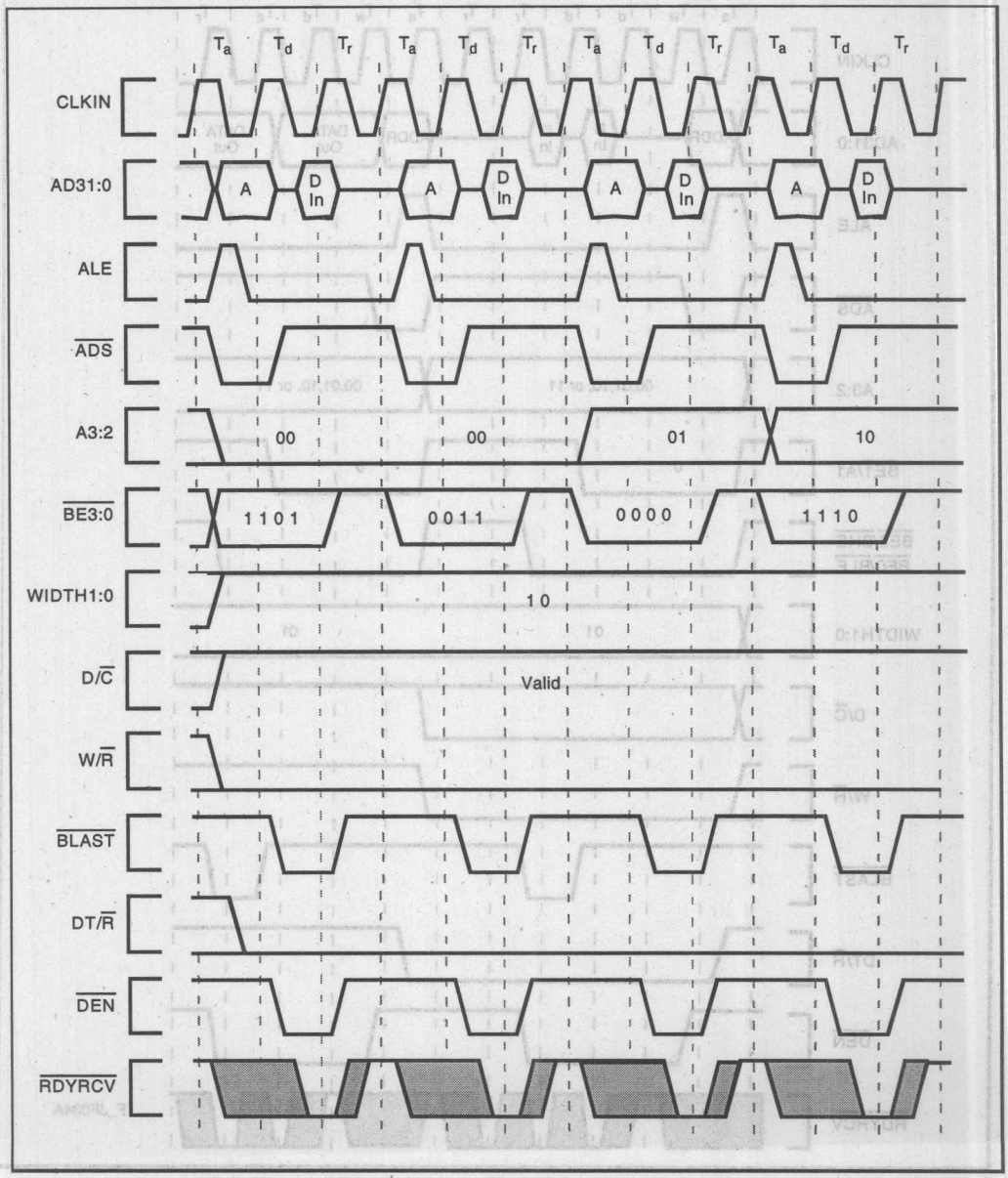
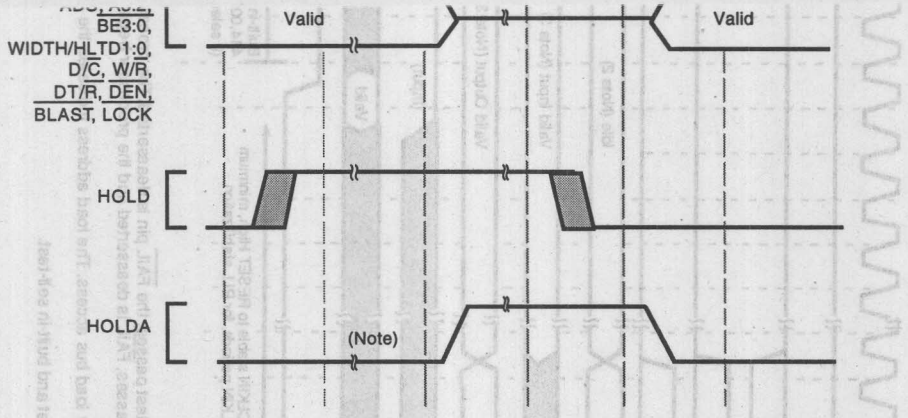
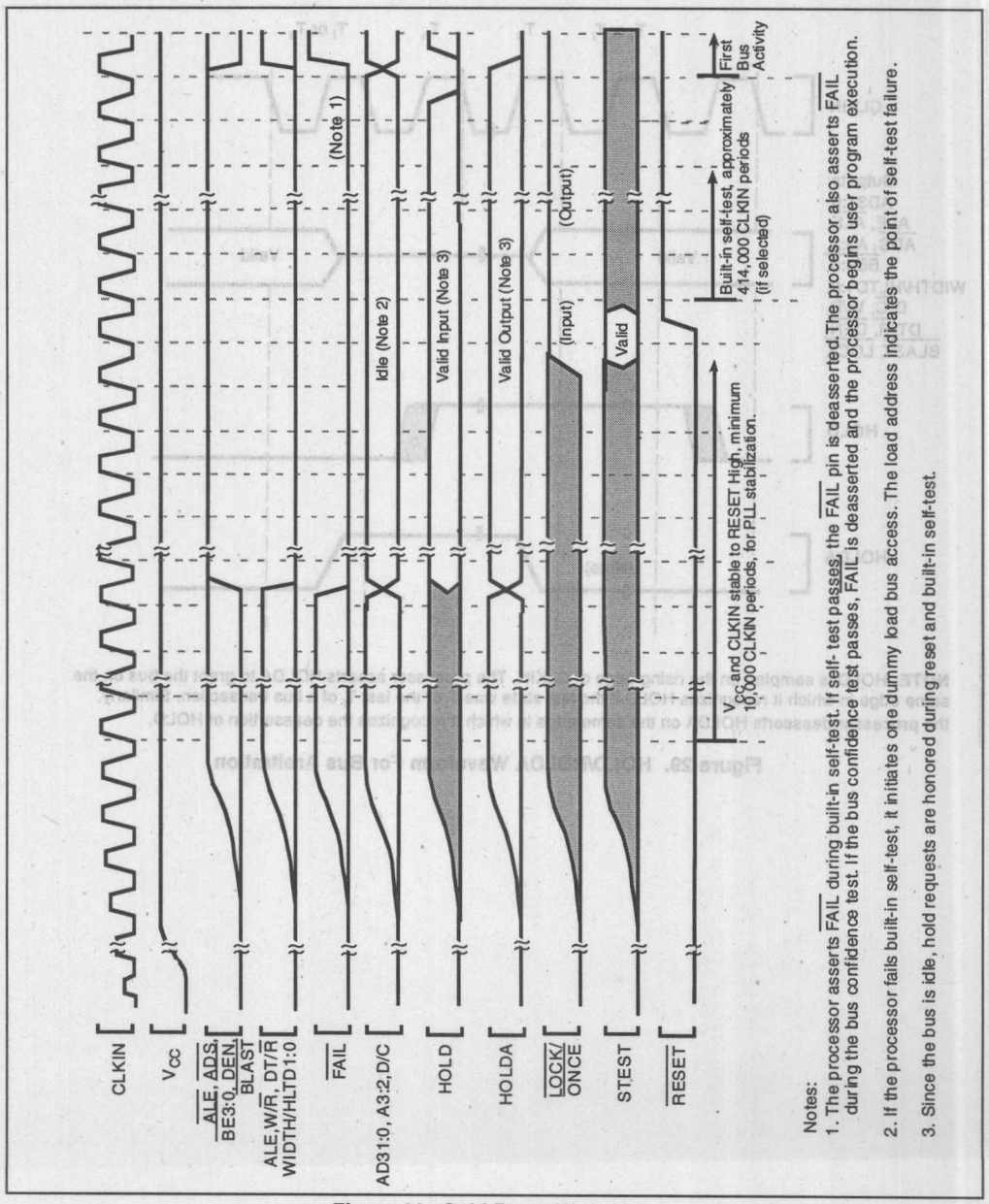


Figure 28. Bus Transactions Generated by Double Word Read Bus Request, Misaligned One Byte From Quad Word Boundary, 32-Bit Bus, Little Endian



NOTE: HOLD is sampled on the rising edge of CLKIN. The processor asserts HOLDA to grant the bus on the same edge in which it recognizes HOLD if the last state was T_i or the last T_r of a bus transaction. Similarly, the processor deasserts HOLDA on the same edge in which it recognizes the deassertion of HOLD.

Figure 29. HOLD/HOLDA Waveform For Bus Arbitration



Notes:
1. The processor asserts FAIL during built-in self-test. If self- test passes, the FAIL pin is deasserted. The processor also asserts FAIL during the bus confidence test. If the bus confidence test passes, FAIL is deasserted and the processor begins user program execution.
2. If the processor fails built-in self-test, it initiates one dummy load bus access. The load address indicates the point of self-test failure.
3. Since the bus is idle, hold requests are honored during reset and built-in self-test.

Figure 30. Cold Reset Waveform

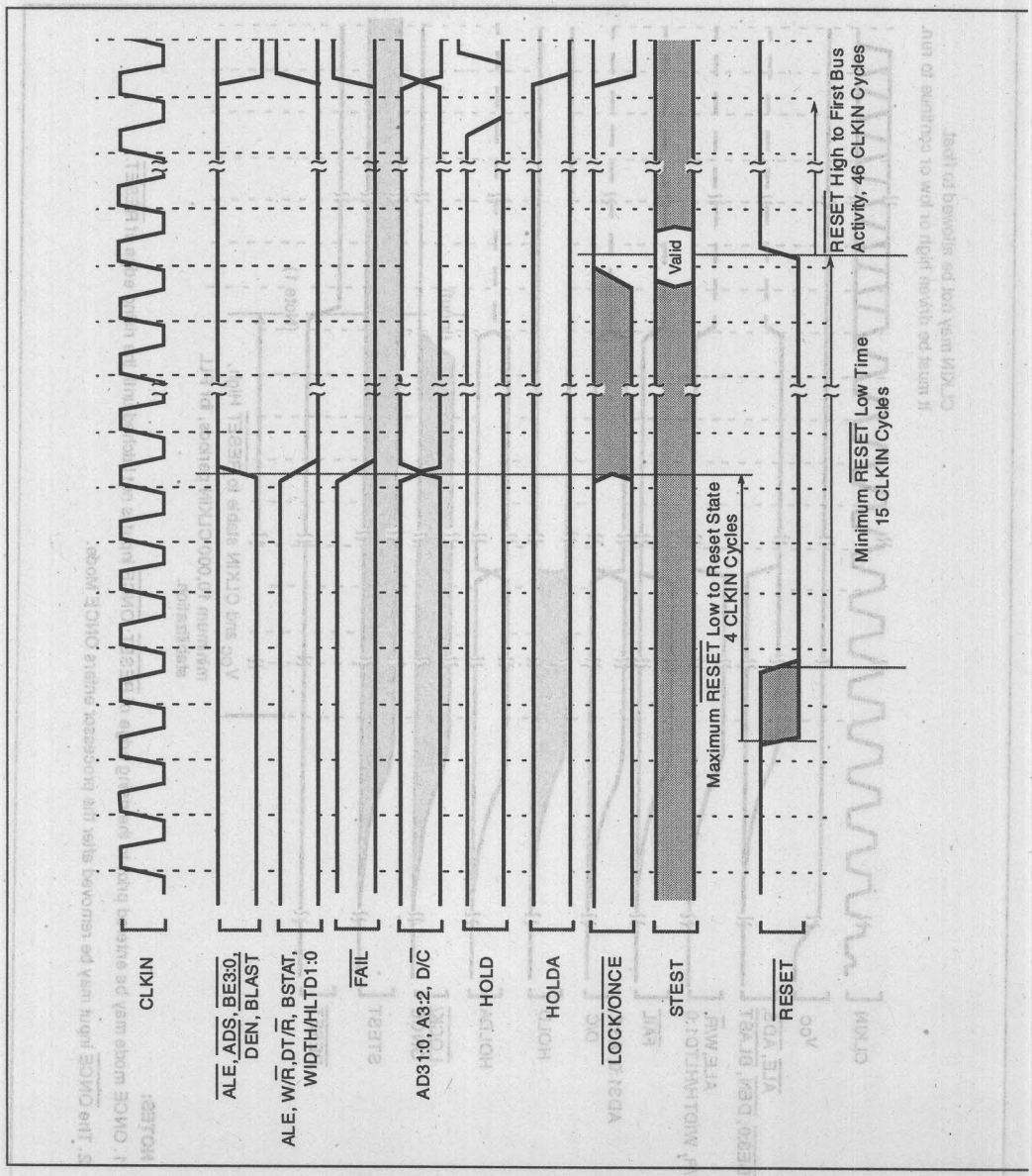


Figure 31. Warm Reset Waveform

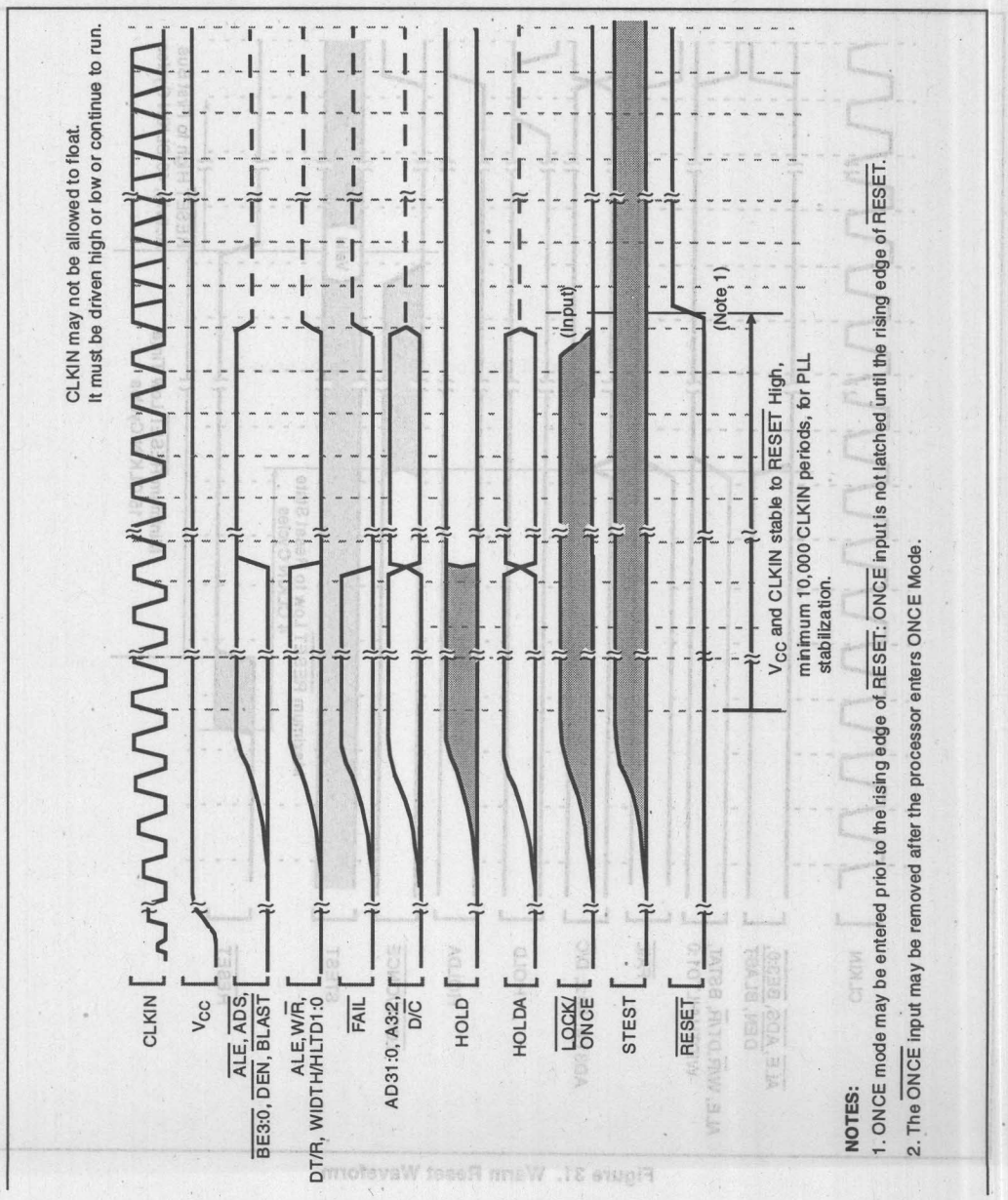


Figure 32. Entering the ONCE State

Quad Word	16

Table 20. Summary of Byte Load and Store Accesses

Address Offset from Natural Boundary (in Bytes)	Accesses on 8-Bit Bus (WIDTH1:0=00)	Accesses on 16-Bit Bus (WIDTH1:0=01)	Accesses on 32-Bit Bus (WIDTH1:0=10)
+0 (aligned)	• byte access	• byte access	• byte access

Table 21. Summary of Short Word Load and Store Accesses

Address Offset from Natural Boundary (in Bytes)	Accesses on 8-Bit Bus (WIDTH1:0=00)	Accesses on 16-Bit Bus (WIDTH1:0=01)	Accesses on 32-Bit Bus (WIDTH1:0=10)
+0 (aligned)	• burst of 2 bytes	• short-word access	• short-word access
+1	• 2 byte accesses	• 2 byte accesses	• 2 byte accesses

Table 22. Summary of *n*-Word Load and Store Accesses (*n* = 1, 2, 3, 4)

Address Offset from Natural Boundary in Bytes	Accesses on 8-Bit Bus (WIDTH1:0=00)	Accesses on 16-Bit Bus (WIDTH1:0=01)	Accesses on 32-Bit Bus (WIDTH1:0=10)
+0 (aligned) (*n* =1, 2, 3, 4)	• *n* burst(s) of 4 bytes	• case *n*=1: burst of 2 short words • case *n*=2: burst of 4 short words • case *n*=3: burst of 4 short words burst of 2 short words • case *n*=4: 2 bursts of 4 short words	• burst of *n* word(s)
+1 (*n* =1, 2, 3, 4) +5 (*n* = 2, 3, 4) +9 (*n* = 3, 4) +13 (*n* = 3, 4)	• byte access • burst of 2 bytes • *n*-1 burst(s) of 4 bytes • byte access	• byte access • short-word access • *n*-1 burst(s) of 2 short words • byte access	• byte access • short-word access • *n*-1 word access(es) • byte access
+2 (*n* =1, 2, 3, 4) +6 (*n* = 2, 3, 4) +10 (*n* = 3, 4) +14 (*n* = 3, 4)	• burst of 2 bytes • *n*-1 burst(s) of 4 bytes • burst of 2 bytes	• short-word access • *n*-1 burst(s) of 2 short words • short-word access	• short-word access • *n*-1 word access(es) • short-word access
+3 (*n* =1, 2, 3, 4) +7 (*n* = 2, 3, 4) +11 (*n* = 3, 4) +15 (*n* = 3, 4)	• byte access • *n*-1 burst(s) of 4 bytes • burst of 2 bytes • byte access	• byte access • *n*-1 burst(s) of 2 short words • short-word access • byte access	• byte access • *n*-1 word access(es) • short-word access • byte access
+4 (*n* = 2, 3, 4) +8 (*n* = 3, 4) +12 (*n* = 3, 4)	• *n* burst(s) of 4 bytes	• *n* burst(s) of 2 short words	• *n* word access(es)

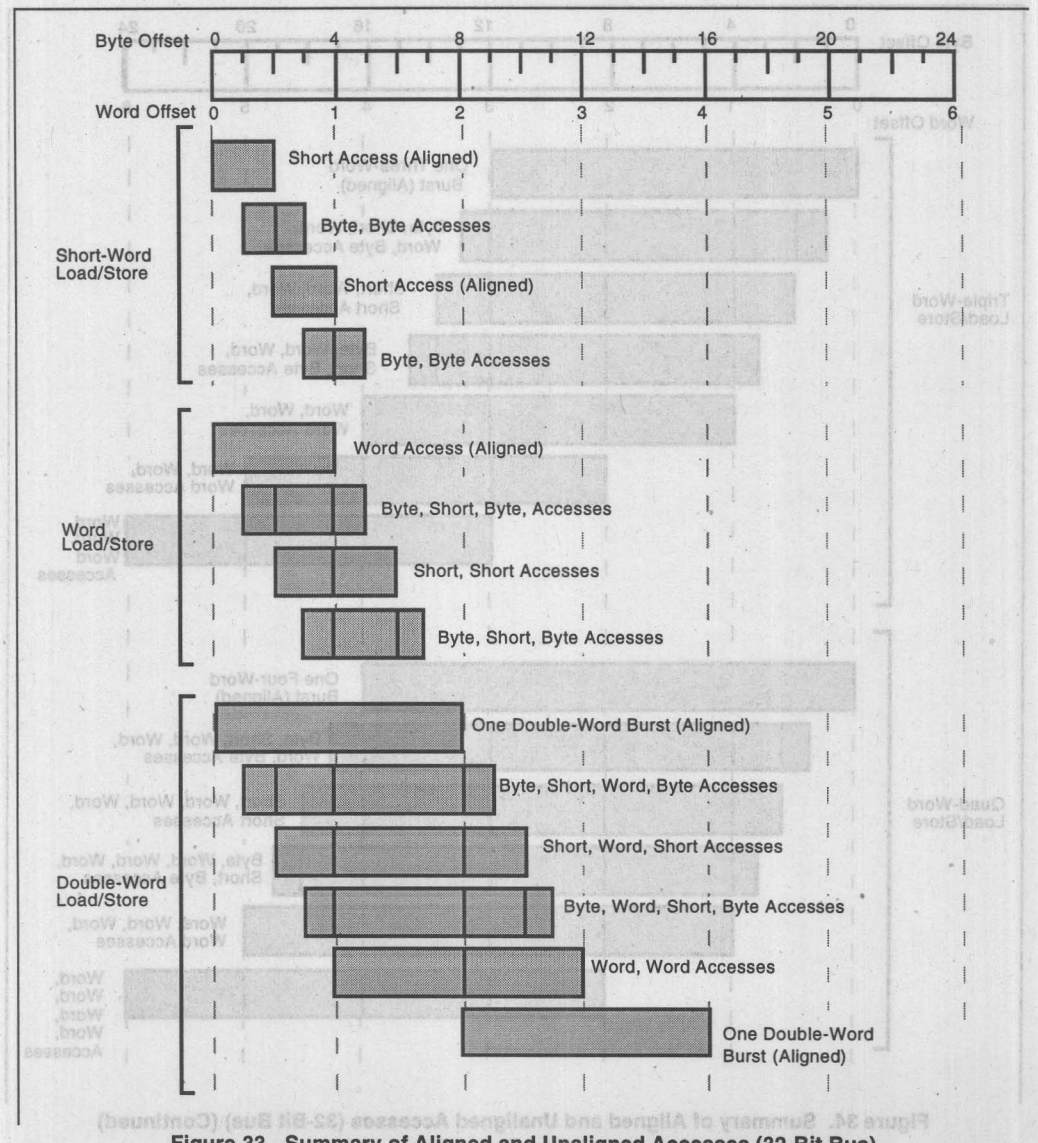


Figure 33. Summary of Aligned and Unaligned Accesses (32-Bit Bus)

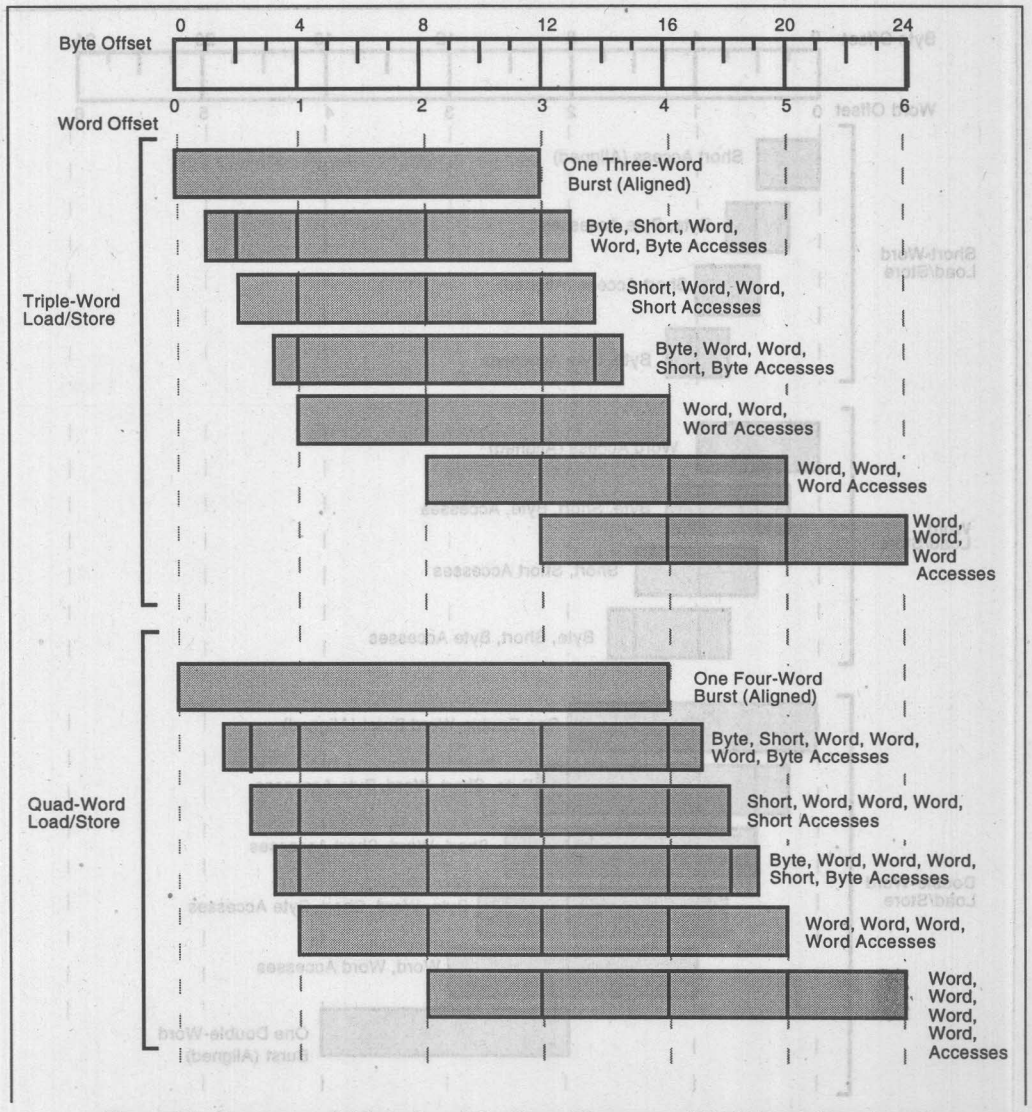


Figure 34. Summary of Aligned and Unaligned Accesses (32-Bit Bus) (Continued)

The device and stepping letter is also printed on the top side of the product package.

Table 23. 80960JA/JF Die and Stepping Reference

Device and Stepping	Version Number	Part Number	Manufacturer	X	Complete ID (Hex)
80960JA A, A2	0000	1000 1000 0010 0001	0000 0001 001	1	08821013
80960JF A, A2	0000	1000 1000 0010 0000	0000 0001 001	1	08820013

NOTE: This data sheet applies to the 80960JA/JF A and 80960JA/JF A2 steppings.

7.0 REVISION HISTORY

This data sheet supersedes revision 272504-003. Table 24 indicates significant changes since that revision. Table 24, Table 24, and Table 24 detail changes which occurred prior to revision 272504-004.

Table 24. Data Sheet Version -003 to -004 Revision History (Sheet 1 of 2)

Section	Description
Table 13, 80960JA/JF DC Characteristics (pg. 23)	Removed Icc characteristics. Added V_{OLP} (output ground bounce) specification.
Table 13, 80960JA/JF DC Characteristics (pg. 23)	New table for comprehensible Icc characteristics. Added Icc's for reset mode. ONCE Icc Improved from 30 mA to 10 mA.
Section 4.5, AC Specifications (pg. 25)	Grouped AC Specifications tables by frequency. Added 25 MHz and 16 MHz AC specifications.
Table 15, 80960JA/JF AC Characteristics (33 MHz) (pg. 25) Section INPUT CLOCK TIMINGS	T_{CS} (max) improved from ±0.1% to ±250 ps
Table 15, 80960JA/JF AC Characteristics (33 MHz) (pg. 25) Section SYNCHRONOUS OUTPUT TIMINGS	T_{ov1} (min) improved from 3.0 ns to 3.5 ns. T_{ov2} (min) improved from $0.5T_C$ + 3.0 ns to $0.5T_C$ + 3.5 ns. T_{oF} (min) improved from 3.0 ns to 3.5 ns.

OUTPUT TIMINGS	
Table 15, 80960JA/JF AC Characteristics (33 MHz) (pg. 25) Section BOUNDARY SCAN TEST SIGNAL TIMINGS	T_{BSF} (max) improved from 8 MHz to $0.5T_F$. T_{BSIS1} (min) improved from 8 ns to 4 ns. T_{BSIH1} (min) improved from 10 ns to 6 ns. T_{BSIS2} (min) improved from 8 ns to 4 ns. T_{BSIH2} (min) improved from 10 ns to 6 ns.

Table 25. Data Sheet Version -002 to -003 Revision History

Section	Description
Section 4.3, Connection Recommendations (pg. 22)	Added precaution to drive $\overline{TRST}$ pin low to reset the TAP controller, even if the JTAG Boundary Scan function will not be used.
Table 13, 80960JA/JF DC Characteristics (pg. 23)	Changed I_{LI} specification to I_{LI1} and added new specification I_{LI2} to denote leakage for TAP controller pins. I_{CC} ONCE mode now indicated as 19mA (typ.) and 30 mA (max.).
Table 15, 80960JA/JF AC Characteristics (33 MHz) (pg. 25)	Minimum CLKIN frequency now specified as 8 MHz.
Table 16, Targeted 80960JA/JF Synchronous Input Timings (pg. 24)	T_{IS1} and T_{IH1} now applicable to $\overline{NMI}$ and $\overline{XINT7:0}$ signals with setup time (min.) improved from 7 to 6 ns and hold time (min.) improved from 3 to 2 ns. T_{IH2} (min.) improved from 2 to 1 ns. T_{IS4} (min.) improved from 7 to 6 ns and T_{IH4} (min.) improved from 3 to 2 ns, with T_{IS4} and T_{IH4} now measured from the inactive edge of $\overline{RESET}$.
Figure 7, Output Delay or Hold vs. Load Capacitance (pg. 32) Figure 8, Rise and Fall Time Derating (pg. 33)	Inserted data.
Figure 12, Input Setup and Hold Waveform for TIS1 and TIH1 (pg. 35) Figure 15, Input Setup and Hold Waveform for TIS4 and TIH4 (pg. 36)	Figures redrawn to correspond to changes in setup and hold specifications for $\overline{NMI}$, $\overline{XINT7:0}$, $\overline{ONCE}$ and STEST.

Table 26. Data Sheet Version -001 to -002 Revision History (Sheet 1 of 2)

Section	Description
Figure 2, 80960JA/JF Block Diagram (pg. 2)	Added 80960JA product version, which includes 2 Kbyte instruction cache and 1 Kbyte data cache.
Table 1, 80960Jx Instruction Set (pg. 5)	Renamed Conditional Move instruction to Conditional Select instruction.
Table 3, Pin Description — External Bus Signals (pg. 7)	Clarified description of unused AD31:0 pins during write operations. Corrected description of AD31:0 pins during HALT mode. Renamed $\overline{ADS}$ pin from Address/Data Status to Address Strobe (pin function unchanged). Corrected description of $\overline{BE3:2}$ pins to indicate high state when pin(s) unused.
Table 4, Pin Description — Processor Control Signals, Test Signals and Power (pg. 10)	Added note that TDO does not float during ONCE Mode. Added note to ground $\overline{TRST}$ if TAP unused.
Figure 3, 132-Lead Pin Grid Array Bottom View - Pins Facing Up (pg. 13) Figure 4, 132-Lead Pin Grid Array Top View - Pins Facing Down (pg. 14) Table 6, 132-Lead PGA Pinout — In Signal Order (pg. 15) Table 7, 132-Lead PGA Pinout — In Pin Order (pg. 16)	New figures and tables. Pin C8 changed from V_{CC} (previously published elsewhere) to NC. Processor operates correctly in a system with either connection.
Figure 5, 132-Lead PQFP - Top View (pg. 17) Table 8, 132-Lead PQFP Pinout — In Signal Order (pg. 18) Table 9, 132-Lead PQFP Pinout — In Pin Order (pg. 19)	New figure and tables. Lead 19 changed from $V_{CC(I/O)}$ (previously published elsewhere) to NC. Processor operates correctly in a system with either connection.
Section 3.2, Package Thermal Specifications (pg. 20)	New section with tables of thermal characteristics.
Section 4.0, ELECTRICAL SPECIFICATIONS (pg. 21)	New section with targeted operating conditions, targeted DC characteristics, targeted AC specifications and AC timing waveforms.
Figure 29, HOLD/HOLDA Waveform For Bus Arbitration (pg. 47)	Improved figure.

Table 26. Data Sheet Version -001 to -002 Revision History (Sheet 2 of 2)

Section	Description
Table 19, Natural Boundaries for Load and Store Accesses (pg. 51) Table 20, Summary of Byte Load and Store Accesses (pg. 51) Table 21, Summary of Short Word Load and Store Accesses (pg. 51) Table 22, Summary of n-Word Load and Store Accesses (n = 1, 2, 3, 4) (pg. 52) Figure 33, Summary of Aligned and Unaligned Accesses (32-Bit Bus) (pg. 53)	New tables and figures to explain bus access alignment.
Figure 30, Cold Reset Waveform (pg. 48)	New figure.
Section 6.0, DEVICE IDENTIFICATION (pg. 55)	New section.

Table 27. Data Sheet Version -002 to -003 Revision History (Sheet 1 of 2)

Section	Description
Section 4.3, Connection Recommendations (pg. 22)	Added precaution to drive $\overline{TRST}$ pin low to reset the TAP controller, even if the JTAG Boundary Scan function will not be used.
Table 13, 80960JA/JF DC Characteristics (pg. 23)	Changed I_{LI} specification to I_{LI1} and added new specification I_{LI2} to denote leakage for TAP controller pins. I_{CC} ONCE mode now indicated as 19mA (typ.) and 30 mA (max.).
Table 15, 80960JA/JF AC Characteristics (33 MHz) (pg. 25)	Minimum CLKIN frequency now specified as 8 MHz.
Table 16, Targeted 80960JA/JF Synchronous Input Timings (pg. 24)	T_{IS1} and T_{IH1} now applicable to $\overline{NMI}$ and $\overline{XINT7:0}$ signals with setup time (min.) improved from 7 to 6 ns and hold time (min.) improved from 3 to 2 ns. T_{IH2} (min.) improved from 2 to 1 ns. T_{IS4} (min.) improved from 7 to 6 ns and T_{IH4} (min.) improved from 3 to 2 ns, with T_{IS4} and T_{IH4} now measured from the inactive edge of $\overline{RESET}$.
Figure 7, Output Delay or Hold vs. Load Capacitance (pg. 32) Figure 8, Rise and Fall Time Derating (pg. 33)	Inserted data.

Table 27. Data Sheet Version -002 to -003 Revision History (Sheet 2 of 2)

Section	Description
Figure 12, Input Setup and Hold Waveform for T_{ISY} and T_{IHY} (pg. 35)	Figures redrawn to correspond to changes in setup and hold specifications for NMI, XINT7:0, $\overline{ONCE}$ and $\overline{STEST}$.
Figure 15, Input Setup and Hold Waveform for T_{ISA} and T_{IHA} (pg. 36)	

PRELIMINARY

80L960JA/JF
3.3 V EMBEDDED 32-BIT MICROPROCESSOR

• 3.3 Volt Version of the 80960JA/JF Processor

- Pin/Code Compatible with all 80960Jx Processors
- High-Performance Embedded Architecture
  - One Instruction/Clock Execution
  - Load/Store Programming Model
  - Sixteen 32-Bit Global Registers
  - Sixteen 32-Bit Local Registers (8 sets)
  - Nine Addressing Modes
  - User/Supervisor Protection Model
- Two-Way Set Associative Instruction Cache
  - 80L960JA - 2 Kbyte
  - 80L960JF - 4 Kbyte
  - Programmable Cache Locking Mechanism
- Direct Mapped Data Cache
  - 80L960JA - 1 Kbyte
  - 80L960JF - 2 Kbyte
  - Write Through Operation
- On-Chip Stack Frame Cache
  - Seven Register Sets Can Be Saved
  - Automatic Allocation on Call/Return
  - 0-7 Frames Reserved for High-Priority Interrupts
- On-Chip Data RAM
  - 1 Kbyte Critical Variable Storage
  - Single-Cycle Access
- High Bandwidth Burst Bus
  - 32-Bit Multiplexed Address/Data
  - Programmable Memory Configuration
  - Selectable 8-, 16-, 32-Bit Bus Widths
  - Supports Unaligned Accesses
  - Big or Little Endian Byte Ordering
- New Instructions
  - Conditional Add, Subtract and Select
  - Processor Management
- High-Speed Interrupt Controller
  - 31 Programmable Priorities
  - Eight Maskable Pins plus NMI
  - Up to 240 Vectors in Expanded Mode
- Two On-Chip Timers
  - Independent 32-Bit Counting
  - Clock Prescaling by 1, 2, 4 or 8
  - Internal Interrupt Sources
- Halt Mode for Low Power
- IEEE 1149.1 (JTAG) Boundary Scan Compatibility
- Packages
  - 132-Lead Pin Grid Array (PGA)
  - 132-Lead Plastic Quad Flat Pack (PQFP)

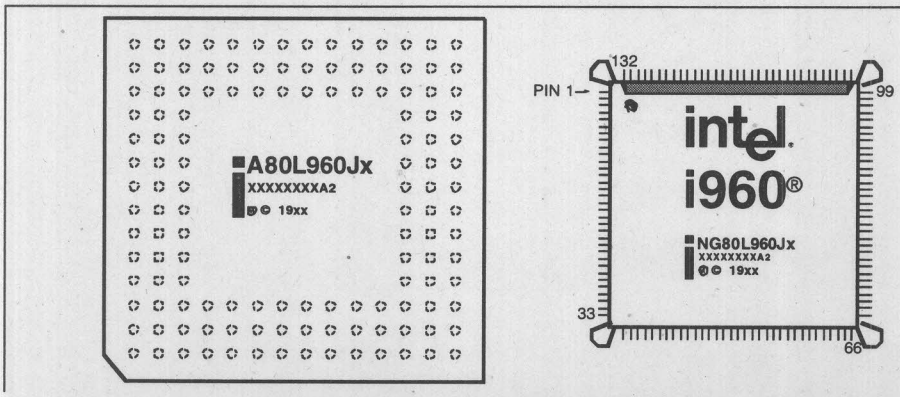


Figure 1. 80L960JA/JF Microprocessor

Order Number: 272744-002

80L960JA/JF
3.3 V EMBEDDED 32-BIT MICROPROCESSOR

- 3.3 Volt Version of the 80960JA/JF Processor

- Pin/Code Compatible with all 80960Jx Processors
- High-Performance Embedded Architecture
  - One Instruction/Clock Execution
  - Load/Store Programming Model
  - Sixteen 32-Bit Global Registers
  - Sixteen 32-Bit Local Registers (8 sets)
  - Nine Addressing Modes
  - User/Supervisor Protection Model
- Two-Way Set Associative Instruction Cache
  - 80L960JA - 2 Kbyte
  - 80L960JF - 4 Kbyte
  - Programmable Cache Locking Mechanism
- Direct Mapped Data Cache
  - 80L960JA - 1 Kbyte
  - 80L960JF - 2 Kbyte
  - Write Through Operation
- On-Chip Stack Frame Cache
  - Seven Register Sets Can Be Saved Automatically on Call/Return
  - 0-7 Frames Reserved for High-Priority Interrupts
- On-Chip Data RAM
  - 1 Kbyte Critical Variable Storage
  - Single-Cycle Access
- High Bandwidth Burst Bus
  - 32-Bit Multiplexed Address/Data
  - Programmable Memory Configuration
  - Selectable 8-, 16-, 32-Bit Bus Widths
  - Supports Unaligned Accesses
  - Big or Little Endian Byte Ordering
- New Instructions
  - Conditional Add, Subtract and Select
  - Processor Management
- High-Speed Interrupt Controller
  - 31 Programmable Priorities
  - Eight Maskable Pins plus NMI
  - Up to 240 Vectors in Expanded Mode
- Two On-Chip Timers
  - Independent 32-Bit Counting
  - Clock Prescaling by 1, 2, 4 or 8
  - Internal Interrupt Sources
- Halt Mode for Low Power
- IEEE 1149.1 (JTAG) Boundary Scan Compatibility
- Packages
  - 132-Lead Pin Grid Array (PGA)
  - 132-Lead Plastic Quad Flat Pack (PQFP)

Figure 1. 80L960JA/JF Microprocessor

Order Number: 272744-002

1.0 PURPOSE

This document contains preliminary information for the 80L960JA/JF microprocessor, including electrical characteristics and package pinout information. Detailed functional descriptions — other than parametric performance — are published in the *i960® Jx Microprocessor User's Guide* (272483).

Throughout this data sheet, references to "80960Jx" indicate features which apply to all of the following:

- 80960JA — 5V, 2 Kbyte instruction cache, 1 Kbyte data cache
- 80960JF — 5V, 4 Kbyte instruction cache, 2 Kbyte data cache
- 80960JD — 5V, 4 Kbyte instruction cache, 2 Kbyte data cache and clock doubling
- 80L960JA — 3.3 V version of the 80960JA
- 80L960JF — 3.3 V version of the 80960JF

2.0 80960Jx OVERVIEW

The 80960Jx offers high performance to cost-sensitive 32-bit embedded applications. The 80960Jx is object code compatible with the 80960 Core Architecture and is capable of sustained execution at the rate of one instruction per clock. This processor's features include generous instruction cache, data cache and data RAM. It also boasts a fast interrupt mechanism, dual programmable timer units and new instructions.

Memory subsystems for cost-sensitive embedded applications often impose substantial wait state penalties. The 80960Jx integrates considerable storage resources on-chip to decouple CPU execution from the external bus.

The 80960Jx rapidly allocates and deallocates local register sets during context switches. The processor needs to flush a register set to the stack only when it saves more than seven sets to its local register cache.

A 32-bit multiplexed burst bus provides a high-speed interface to system memory and I/O. A full complement of control signals simplifies the connection of the 80960Jx to external components. The user programs physical and logical memory attributes through memory-mapped control registers (MMRs) — an extension not found on the i960 Kx, Sx or Cx processors. Physical and logical configuration registers enable the processor to operate with all combinations of bus width and data object alignment. The processor supports a homogeneous byte ordering model.

This processor integrates two important peripherals: a timer unit and an interrupt controller. These and other hardware resources are programmed through memory-mapped control registers, an extension to the familiar 80960 architecture.

The timer unit (TU) offers two independent 32-bit timers for use as real-time system clocks and general-purpose system timing. These operate in either single-shot or auto-reload mode and can generate interrupts.

The interrupt controller unit (ICU) provides a flexible, low-latency means for requesting interrupts.The ICU provides full programmability of up to 240 interrupt sources into 31 priority levels. The ICU takes advantage of a cached priority table and optional routine caching to minimize interrupt latency. Local registers may be dedicated to high-priority interrupts to further reduce latency. Acting independently from the core, the ICU compares the priorities of posted interrupts with the current process priority, off-loading this task from the core. The ICU also supports the integrated timer interrupts.

The 80960Jx features a Halt mode designed to support applications where low power consumption is critical. The **halt** instruction shuts down instruction execution, resulting in a power savings of up to 90 percent.

The 80960Jx's testability features, including ONCE (On-Circuit Emulation) mode and Boundary Scan (JTAG), provide a powerful environment for design debug and fault diagnosis.

The *Solutions960®* program features a wide variety of development tools which support the i960 processor family. Many of these tools are developed by partner companies; some are developed by Intel, such as profile-driven optimizing compilers. For more information on these products, contact your local Intel representative.

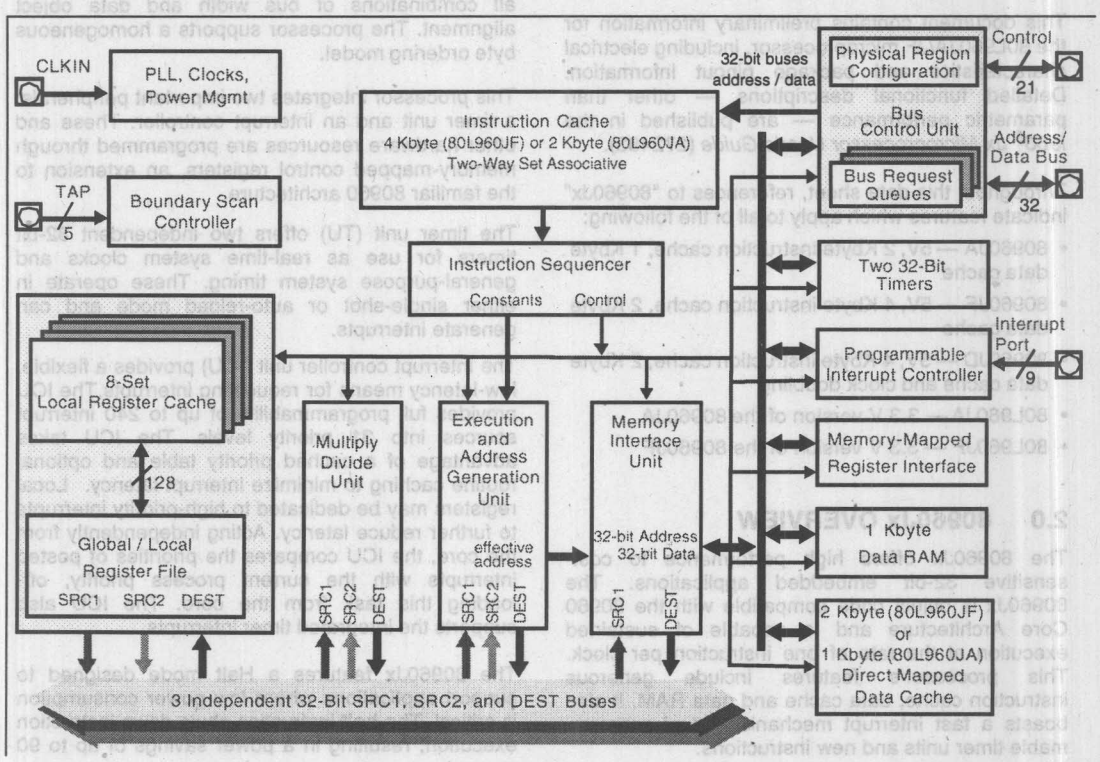


Figure 2. 80960Jx Block Diagram

2.1 80960 Processor Core

The 80960Jx family is a scalar implementation of the 80960 Core Architecture. Intel designed this processor core as a very high performance device that is also cost-effective. Factors that contribute to the core's performance include:

- Single-clock execution of most instructions
- Independent Multiply/Divide Unit
- Efficient instruction pipeline minimizes pipeline break latency
- Register and resource scoreboarding allow overlapped instruction execution
- 128-bit register bus speeds local register caching
- Two-way set associative, integrated instruction cache
- Direct-mapped, integrated data cache
- 1 Kbyte integrated data RAM delivers zero wait state program data

2.2 Burst Bus

A 32-bit high-performance bus controller interfaces the 80960Jx to external memory and peripherals. The BCU fetches instructions and transfers data at the rate of up to four 32-bit words per six clock cycles. The external address/data bus is multiplexed.

Users may configure the 80960Jx's bus controller to match an application's fundamental memory organization. Physical bus width is register-programmed for up to eight regions. Byte ordering and data caching are programmed through a group of logical memory templates and a defaults register.

The BCU's features include:

- Multiplexed external bus to minimize pin count
- 32-, 16- and 8-bit bus widths to simplify I/O interfaces
- External ready control for address-to-data, data-to-data and data-to-next-address wait state types
- Support for big or little endian byte ordering to facilitate the porting of existing program code
- Unaligned bus accesses performed transparently
- Three-deep load/store queue to decouple the bus from the core

Upon reset, the 80960Jx conducts an internal self test. Then, before executing its first instruction, it performs an external bus confidence test by performing a checksum on the first words of the initialization boot record (IBR).

The user may examine the contents of the caches at any time by executing special cache control instructions.

2.3 Timer Unit

The timer unit (TU) contains two independent 32-bit timers which are capable of counting at several clock rates and generating interrupts. Each is programmed by use of the TU registers. These memory-mapped registers are addressable on 32-bit boundaries. The timers have a single-shot mode and auto-reload capabilities for continuous operation. Each timer has an independent interrupt request to the 80960Jx's interrupt controller. The TU can generate a fault when unauthorized writes from user mode are detected. Clock prescaling is supported.

2.4 Priority Interrupt Controller

A programmable interrupt controller manages up to 240 external sources through an 8-bit external interrupt port. Alternatively, the interrupt inputs may be configured for individual edge- or level-triggered inputs. The interrupt unit (IU) also accepts interrupts from the two on-chip timer channels and a single Non-Maskable Interrupt (NMI) pin. Interrupts are serviced according to their priority levels relative to the current process priority.

Low interrupt latency is critical to many embedded applications. As part of its highly flexible interrupt mechanism, the 80960Jx exploits several techniques to minimize latency:

- Interrupt vectors and interrupt handler routines can be reserved on-chip
- Register frames for high-priority interrupt handlers can be cached on-chip
- The interrupt stack can be placed in cacheable memory space

2.5 Instruction Set Summary

The 80960Jx adds several new instructions to the i960 core architecture. The new instructions are:

- Conditional Move
- Conditional Add
- Conditional Subtract
- Byte Swap
- Halt
- Cache Control
- Interrupt Control

Table 1 identifies the instructions that the 80960Jx supports. Refer to *i960® Jx Microprocessor User's Guide* (272483) for a detailed description of each instruction.

2.6 Faults and Debugging

The 80960Jx employs a comprehensive fault model. The processor responds to faults by making implicit calls to a fault handling routine. Specific information collected for each fault allows the fault handler to diagnose exceptions and recover appropriately.

The processor also has built-in debug capabilities. In software, the 80960Jx may be configured to detect as many as seven different trace event types. Alternatively, **mark** and **fmark** instructions can generate

trace events explicitly in the instruction stream. Hardware breakpoint registers are also available to trap on execution and data addresses.

2.7 Low Power Operation

Intel fabricates the 80960Jx using an advanced sub-micron manufacturing process. The processor's sub-micron topology provides the circuit density for optimal cache size and high operating speeds while dissipating modest power. The processor also uses dynamic power management to turn off clocks to unused circuits.

Users may program the 80960Jx to enter Halt mode for maximum power savings. In Halt mode, the processor core stops completely while the integrated peripherals continue to function, reducing overall power requirements up to 90 percent. Processor execution resumes from internally or externally generated interrupts.

2.8 Test Features

The 80960Jx incorporates numerous features which enhance the user's ability to test both the processor and the system to which it is attached. These features include ONCE (On-Circuit Emulation) mode and Boundary Scan (JTAG).

The 80960Jx provides testability features compatible with IEEE Standard Test Access Port and Boundary Scan Architecture (IEEE Std. 1149.1).

One of the boundary scan instructions, HIGHZ, forces the processor to float all its output pins (ONCE mode). ONCE mode can also be initiated at reset without using the boundary scan mechanism.

ONCE mode is useful for board-level testing. This feature allows a mounted 80960Jx to electrically "remove" itself from a circuit board. This allows for system-level testing where a remote tester — such as an in-circuit emulator — can exercise the processor system.

The provided test logic does not interfere with component or circuit board behavior and ensures that components function correctly, connections between various components are correct, and various components interact correctly on the printed circuit board.

The JTAG Boundary Scan feature is an attractive alternative to conventional "bed-of-nails" testing. It can examine connections which might otherwise be inaccessible to a test system.

2.9 Memory-Mapped Control Registers

The 80960Jx, though compliant with i960 series processor core, has the added advantage of memory-mapped, internal control registers not found on the i960 Kx, Sx or Cx processors. These give software the interface to easily read and modify internal control registers.

Each of these registers is accessed as a memory-mapped, 32-bit register. Access is accomplished through regular memory-format instructions. The processor ensures that these accesses do not generate external bus cycles.

2.10 Data Types and Memory Addressing Modes

As with all i960 family processors, the 80960Jx instruction set supports several data types and formats:

- Bit
- Bit fields
- Integer (8-, 16-, 32-, 64-bit)
- Ordinal (8-, 16-, 32-, 64-bit unsigned integers)
- Triple word (96 bits)
- Quad word (128 bits)

The 80960Jx provides a full set of addressing modes for C and assembly programming:

- Two Absolute modes
- Five Register Indirect modes
- Index with displacement
- IP with displacement

Table 1. 80960Jx Instruction Set

Data Movement	Arithmetic	Logical	Bit, Bit Field and Byte
Load	Add	And	Set Bit
Store	Subtract	Not And	Clear Bit
Move	Multiply	And Not	Not Bit
*Conditional Select	Divide	Or	Alter Bit
Load Address	Remainder	Exclusive Or	Scan For Bit
	Modulo	Not Or	Span Over Bit
	Shift	Or Not	Extract
	Extended Shift	Nor	Modify
	Extended Multiply	Exclusive Nor	Scan Byte for Equal
	Extended Divide	Not	*Byte Swap
	Add with Carry	Nand	
	Subtract with Carry		
	*Conditional Add		
	*Conditional Subtract		
	Rotate		
Comparison	**Branch**	**Call/Return**	**Fault**
Compare	Unconditional Branch	Call	Conditional Fault
Conditional Compare	Conditional Branch	Call Extended	Synchronize Faults
Compare and Increment	Compare and Branch	Call System	
Compare and Decrement		Return	
Test Condition Code		Branch and Link	
Check Bit			
Debug	**Processor Management**	**Atomic**	
Modify Trace Controls	Flush Local Registers	Atomic Add	
Mark	Modify Arithmetic Controls	Atomic Modify	
Force Mark	Modify Process Controls		
	*Halt		
	System Control		
	*Cache Control		
	*Interrupt Control		

NOTE: Asterisk (*) denotes new 80960Jx instructions unavailable on 80960CA/CF, 80960KA/KB and 80960SA/SB implementations.

3.0 PACKAGE INFORMATION

The 80L960JA/JF will be offered in several speed and package types. The 132-pin Pin Grid Array (PGA) device will be specified for operation at V_{CC} = 3.3 V ± 5% over a case temperature range of 0° to 100°C:

- A80L960JA/JF-25 (25 MHz)
- A80L960JA/JF-16 (16 MHz)

The 132-pin Plastic Quad Flatpack (PQFP) devices will be specified for operation at V_{CC} = 3.3 V ± 5% over a case temperature range of 0° to 100°C:

- NG80L960JA/JF-25 (25 MHz)
- NG80L960JA/JF-16 (16 MHz)

For complete package specifications and information, refer to Intel's Packaging Handbook (240800).

3.1 Pin Descriptions

This section describes the pins for the 80L960JA/JF in the 132-pin ceramic Pin Grid Array (PGA) package and 132-lead Plastic Quad Flatpack Package (PQFP).

Section 3.1.1, Functional Pin Definitions describes pin function; **Section 3.1.2, 80960Jx 132-Lead PGA Pinout** and **Section 3.1.3, 80960Jx PQFP Pinout** define the signal and pin locations for the supported package types.

3.1.1 Functional Pin Definitions

Table 2 presents the legend for interpreting the pin descriptions which follow. Pins associated with the bus interface are described in Table 3. Pins associated with basic control and test functions are described in Table 4. Pins associated with the Interrupt Unit are described in Table 5.

Table 2. Pin Description Nomenclature

Symbol	Description
I	Input pin only.
O	Output pin only.
I/O	Pin can be either an input or output.
–	Pin must be connected as described.
S	Synchronous. Inputs must meet setup and hold times relative to CLKIN for proper operation. S(E) Edge sensitive input S(L) Level sensitive input
A (...)	Asynchronous. Inputs may be asynchronous relative to CLKIN. A(E) Edge sensitive input A(L) Level sensitive input
R (...)	While the processor's $\overline{RESET}$ pin is asserted, the pin: R(1) is driven to V_{CC} R(0) is driven to V_{SS} R(Q) is a valid output R(X) is driven to unknown state R(H) is pulled up to V_{CC}
H (...)	While the processor is in the hold state, the pin: H(1) is driven to V_{CC} H(0) is driven to V_{SS} H(Q) Maintains previous state or continues to be a valid output H(Z) Floats
P (...)	While the processor is halted, the pin: P(1) is driven to V_{CC} P(0) is driven to V_{SS} P(Q) Maintains previous state or continues to be a valid output

	P(Q)	AD15:8 and AD7:0. During write operations, unused pins are driven to determinate values. SIZE, which comprises bits 0-1 of the AD lines during a T_a cycle, specifies the number of data transfers during the bus transaction. AD1 AD0 Bus Transfers 0 0 1 Transfer 0 1 2 Transfers 1 0 3 Transfers 1 1 4 Transfers When the processor enters Halt mode, if the previous bus operation was a: • write — AD31:2 are driven with the last data value on the AD bus. • read — AD31:4 are driven with the last address value on the AD bus; AD3:2 are driven with the value of A3:2 from the last data cycle. Typically, AD1:0 reflect the SIZE information of the last bus transaction (either instruction fetch or load/store) that was executed before entering Halt mode.
ALE	**O** R(0) H(Z) P(0)	**ADDRESS LATCH ENABLE** indicates the transfer of a physical address. ALE is asserted during a T_a cycle and deasserted before the beginning of the T_d state. It is active HIGH and floats to a high impedance state during a hold cycle (T_h).
$\overline{\text{ALE}}$	**O** R(1) H(Z) P(1)	**ADDRESS LATCH ENABLE** indicates the transfer of a physical address. $\overline{\text{ALE}}$ is the inverted version of ALE. This signal gives the 80960Jx a high degree of compatibility with existing 80960Kx systems.
$\overline{\text{ADS}}$	**O** R(1) H(Z) P(1)	**ADDRESS STROBE** indicates a valid address and the start of a new bus access. The processor asserts $\overline{\text{ADS}}$ for the entire T_a cycle. External bus control logic typically samples $\overline{\text{ADS}}$ at the end of the cycle.
A3:2	**O** R(X) H(Z) P(Q)	**ADDRESS3:2** comprise a partial demultiplexed address bus. *32-bit memory accesses:* the processor asserts address bits A3:2 during T_a. The partial word address increments with each assertion of $\overline{\text{RDYRCV}}$ during a burst. *16-bit memory accesses:* the processor asserts address bits A3:1 during T_a with A1 driven on the $\overline{\text{BE1}}$ pin. The partial short word address increments with each assertion of $\overline{\text{RDYRCV}}$ during a burst. *8-bit memory accesses:* the processor asserts address bits A3:0 during T_a, with A1:0 driven on $\overline{\text{BE1:0}}$. The partial byte address increments with each assertion of $\overline{\text{RDYRCV}}$ during a burst.

Table 3. Pin Description — External Bus Signals (Sheet 2 of 4)

NAME	TYPE	DESCRIPTION
$\overline{BE3:0}$	**O** R(1) H(Z) P(1)	**BYTE ENABLES** select which of up to four data bytes on the bus participate in the current bus access. Byte enable encoding is dependent on the bus width of the memory region accessed: *32-bit bus:* $\overline{BE3}$ enables data on AD31:24 $\overline{BE2}$ enables data on AD23:16 $\overline{BE1}$ enables data on AD15:8 $\overline{BE0}$ enables data on AD7:0 *16-bit bus:* $\overline{BE3}$ becomes Byte High Enable (enables data on AD15:8) $\overline{BE2}$ is not used (state is high) $\overline{BE1}$ becomes Address Bit 1 (A1) $\overline{BE0}$ becomes Byte Low Enable (enables data on AD7:0) *8-bit bus:* $\overline{BE3}$ is not used (state is high) $\overline{BE2}$ is not used (state is high) $\overline{BE1}$ becomes Address Bit 1 (A1) $\overline{BE0}$ becomes Address Bit 0 (A0) The processor asserts byte enables, byte high enable and byte low enable during T_a. Since unaligned bus requests are split into separate bus transactions, these signals do not toggle during a burst. They remain active through the last T_d cycle. For accesses to 8- and 16-bit memory, the processor asserts the address bits in conjunction with A3:2 described above.
WIDTH/ HLTD1:0	**O** R(0) H(Z) P(1)	**WIDTH/HALTED** signals denote the physical memory attributes for a bus transaction: WIDTH/HLTD1 — WIDTH/HLTD0 0 — 0 — 8 Bits Wide 0 — 1 — 16 Bits Wide 1 — 0 — 32 Bits Wide 1 — 1 — Processor Halted The processor floats the WIDTH/HLTD pins whenever it relinquishes the bus in response to a HOLD request, regardless of prior operating state.
D/$\overline{C}$	**O** R(X) H(Z) P(Q)	**DATA/CODE** indicates that a bus access is a data access (1) or an instruction access (0). D/$\overline{C}$ has the same timing as W/$\overline{R}$. 0 = instruction access 1 = data access
W/$\overline{R}$	**O** R(0) H(Z) P(Q)	**WRITE/READ** specifies, during a T_a cycle, whether the operation is a write (1) or read (0). It is latched on-chip and remains valid during T_d cycles. 0 = read 1 = write

Table 3. Pin Description — External Bus Signals (Sheet 3 of 4)

NAME	TYPE	DESCRIPTION
DT/$\overline{R}$	O R(0) H(Z) P(Q)	**DATA TRANSMIT / RECEIVE** indicates the direction of data transfer to and from the address/data bus. It is low during T_a and T_w/T_d cycles for a read; it is high during T_a and T_w/T_d cycles for a write. DT/$\overline{R}$ never changes state when $\overline{DEN}$ is asserted. 0 = receive 1 = transmit
$\overline{DEN}$	O R(1) H(Z) P(1)	**DATA ENABLE** indicates data transfer cycles during a bus access. $\overline{DEN}$ is asserted at the start of the first data cycle in a bus access and deasserted at the end of the last data cycle. $\overline{DEN}$ is used with DT/$\overline{R}$ to provide control for data transceivers connected to the data bus. 0 = data cycle 1 = not data cycle
$\overline{BLAST}$	O R(1) H(Z) P(1)	**BURST LAST** indicates the last transfer in a bus access. $\overline{BLAST}$ is asserted in the last data transfer of burst and non-burst accesses. $\overline{BLAST}$ remains active as long as wait states are inserted via the $\overline{RDYRCV}$ pin. $\overline{BLAST}$ becomes inactive after the final data transfer in a bus cycle. 0 = last data transfer 1 = not last data transfer
$\overline{RDYRCV}$	I S(L)	**READY/RECOVER** indicates that data on AD lines can be sampled or removed. If $\overline{RDYRCV}$ is not asserted during a T_d cycle, the T_d cycle is extended to the next cycle by inserting a wait state (T_w). 0 = sample data 1 = don't sample data The $\overline{RDYRCV}$ pin has another function during the recovery (T_r) state. The processor continues to insert additional recovery states until it samples the pin HIGH. This function gives slow external devices more time to float their buffers before the processor begins to drive address again. 0 = insert wait states 1 = recovery complete
$\overline{LOCK}$/ ONCE	I/O S(L) R(H) H(Z) P(1)	**BUS LOCK** indicates that an atomic read-modify-write operation is in progress. The $\overline{LOCK}$ output is asserted in the first clock of an atomic operation and deasserted in the last data transfer of the sequence. The processor does not grant HOLDA while it is asserting $\overline{LOCK}$. This prevents external agents from accessing memory involved in semaphore operations. 0 = Atomic read-modify-write in progress 1 = Atomic read-modify-write not in progress **ONCE MODE:** The processor samples the $\overline{ONCE}$ input during reset. If it is asserted LOW at the end of reset, the processor enters ONCE mode. In ONCE mode, the processor stops all clocks and floats all output pins. The pin has a weak internal pullup which is active during reset to ensure normal operation when the pin is left unconnected. 0 = ONCE mode enabled 1 = ONCE mode not enabled

Table 3. Pin Description — External Bus Signals (Sheet 4 of 4)

NAME	TYPE	DESCRIPTION
HOLD	I S(L)	**HOLD**: A request from an external bus master to acquire the bus. When the processor receives HOLD and grants bus control to another master, it asserts HOLDA, floats the address/data and control lines and enters the T_h state. When HOLD is deasserted, the processor deasserts HOLDA and enters either the T_i or T_a state, resuming control of the address/data and control lines. 0 = no hold request 1 = hold request
HOLDA	O R(Q) H(1) P(Q)	**HOLD ACKNOWLEDGE** indicates to an external bus master that the processor has relinquished control of the bus. The processor can grant HOLD requests and enter the T_h state during reset and while halted as well as during regular operation. 0 = hold not acknowledged 1 = hold acknowledged
BSTAT	O R(0) H(Q) P(0)	**BUS STATUS** indicates that the processor may soon stall unless it has sufficient access to the bus; see *i960® Jx Microprocessor User's Guide* (272483). Arbitration logic can examine this signal to determine when an external bus master should acquire/relinquish the bus. 0 = no potential stall 1 = potential stall

Table 4. Pin Description — Processor Control Signals, Test Signals and Power (Sheet 1 of 2)

NAME	TYPE	DESCRIPTION
CLKIN	I	**CLOCK INPUT** provides the processor's fundamental time base; both the processor core and the external bus run at the CLKIN rate. All input and output timings are specified relative to a rising CLKIN edge.
$\overline{\text{RESET}}$	I A(L)	**RESET** initializes the processor and clears its internal logic. During reset, the processor places the address/data bus and control output pins in their idle (inactive) states. During reset, the input pins are ignored with the exception of $\overline{\text{LOCK/ONCE}}$, STEST and HOLD. The $\overline{\text{RESET}}$ pin has an internal synchronizer. To ensure predictable processor initialization during power up, $\overline{\text{RESET}}$ must be asserted a minimum of 10,000 CLKIN cycles with V_{CC} and CLKIN stable. On a warm reset, $\overline{\text{RESET}}$ should be asserted for a minimum of 15 cycles.
STEST	I S(L)	**SELF TEST** enables or disables the processor's internal self-test feature at initialization. STEST is examined at the end of reset. When STEST is asserted, the processor performs its internal self-test and the external bus confidence test. When STEST is deasserted, the processor performs only the external bus confidence test. 0 = self test disabled 1 = self test enabled

		0 = self test failed 1 = self test passed
TCK	I	**TEST CLOCK** is a CPU input which provides the clocking function for IEEE 1149.1 Boundary Scan Testing (JTAG). State information and data are clocked into the processor on the rising edge; data is clocked out of the processor on the falling edge.
TDI	I S(L)	**TEST DATA INPUT** is the serial input pin for JTAG. TDI is sampled on the rising edge of TCK, during the SHIFT-IR and SHIFT-DR states of the Test Access Port.
TDO	O R(Q) HQ) P(Q)	**TEST DATA OUTPUT** is the serial output pin for JTAG. TDO is driven on the falling edge of TCK during the SHIFT-IR and SHIFT-DR states of the Test Access Port. At other times, TDO floats. TDO does not float during ONCE mode.
$\overline{\text{TRST}}$	I A(L)	**TEST RESET** asynchronously resets the Test Access Port (TAP) controller function of IEEE 1149.1 Boundary Scan testing (JTAG). When using the Boundary Scan feature, connect a pulldown resistor between this pin and V_{SS}. If TAP is not used, this pin must be connected to V_{SS}; however, no resistor is required. See Section 4.3, Connection Recommendations (pg. 23).
TMS	I S(L)	**TEST MODE SELECT** is sampled at the rising edge of TCK to select the operation of the test logic for IEEE 1149.1 Boundary Scan testing.
V_{CC}	–	**POWER** pins intended for external connection to a V_{CC} board plane.
V_{CCPLL}	–	**PLL POWER** is a separate V_{CC} supply pin for the phase lock loop clock generator. It is intended for external connection to the V_{CC} board plane. In noisy environments, add a simple bypass filter circuit to reduce noise-induced clock jitter and its effects on timing relationships.
V_{SS}	–	**GROUND** pins intended for external connection to a V_{SS} board plane.
NC	–	**NO CONNECT** pins. Do not make any system connections to these pins.

Table 5. Pin Description — Interrupt Unit Signals

NAME	TYPE	DESCRIPTION
$\overline{\text{XINT7:0}}$	I A(E/L)	**EXTERNAL INTERRUPT** pins are used to request interrupt service. The $\overline{\text{XINT7:0}}$ pins can be configured in three modes: **Dedicated Mode**: Each pin is assigned a dedicated interrupt level. Dedicated inputs can be programmed to be level (low) or edge (falling) sensitive. **Expanded Mode**: All eight pins act as a vectored interrupt source. The interrupt pins are level sensitive in this mode. **Mixed Mode**: The $\overline{\text{XINT7:5}}$ pins act as dedicated sources and the $\overline{\text{XINT4:0}}$ pins act as the five most significant bits of a vectored source. The least significant bits of the vectored source are set to 010_2 internally. Unused external interrupt pins should be connected to V_{CC}.
$\overline{\text{NMI}}$	I A(E)	**NON-MASKABLE INTERRUPT** causes a non-maskable interrupt event to occur. NMI is the highest priority interrupt source and is falling edge-triggered. If NMI is unused, it should be connected to V_{CC}.

3.1.2 80960Jx 132-Lead PGA Pinout

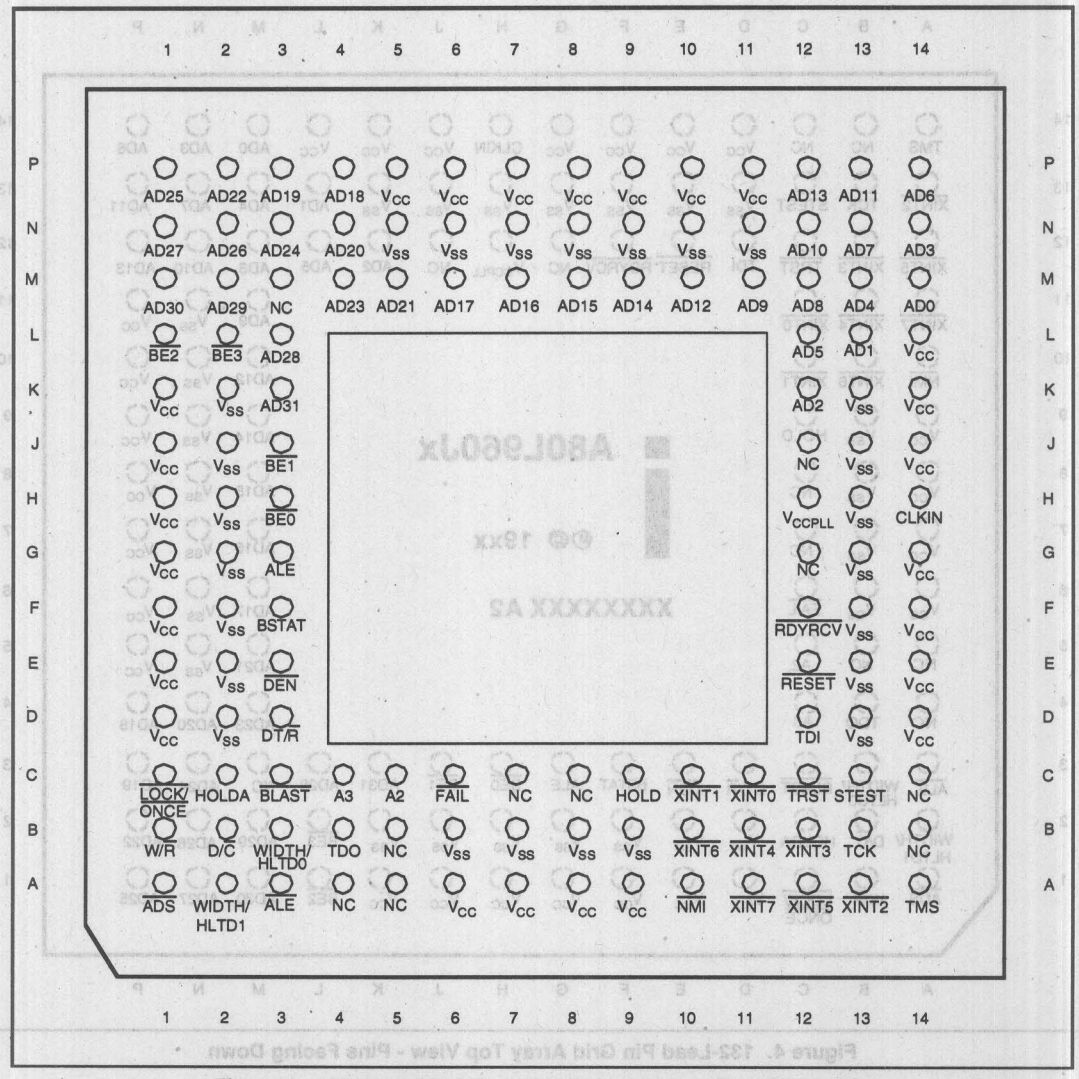


Figure 3. 132-Lead Pin Grid Array Bottom View - Pins Facing Up

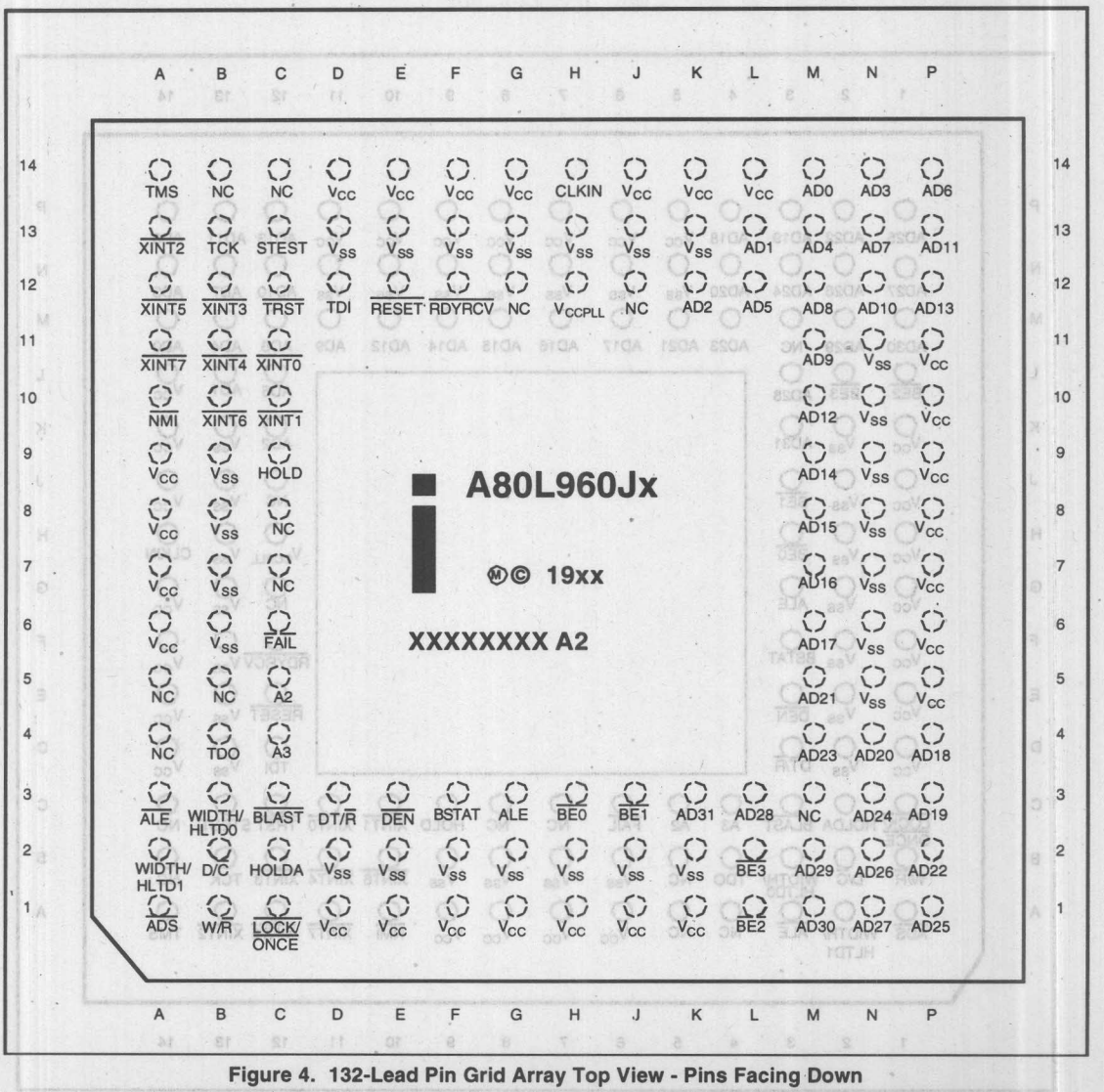


Figure 4. 132-Lead Pin Grid Array Top View - Pins Facing Down

AD2	K12	$\overline{BE0}$	H3	V_{CC}	A6	V_{SS}	E13
AD3	N14	$\overline{BE1}$	J3	V_{CC}	A7	V_{SS}	F2
AD4	M13	$\overline{BE2}$	L1	V_{CC}	A8	V_{SS}	F13
AD5	L12	$\overline{BE3}$	L2	V_{CC}	A9	V_{SS}	G2
AD6	P14	$\overline{BLAST}$	C3	V_{CC}	D1	V_{SS}	G13
AD7	N13	BSTAT	F3	V_{CC}	D14	V_{SS}	H2
AD8	M12	CLKIN	H14	V_{CC}	E1	V_{SS}	H13
AD9	M11	D/$\overline{C}$	B2	V_{CC}	E14	V_{SS}	J2
AD10	N12	$\overline{DEN}$	E3	V_{CC}	F1	V_{SS}	J13
AD11	P13	DT/$\overline{R}$	D3	V_{CC}	F14	V_{SS}	K2
AD12	M10	$\overline{FAIL}$	C6	V_{CC}	G1	V_{SS}	K13
AD13	P12	HOLD	C9	V_{CC}	G14	V_{SS}	N5
AD14	M9	HOLDA	C2	V_{CC}	H1	V_{SS}	N6
AD15	M8	$\overline{LOCK/ONCE}$	C1	V_{CC}	J1	V_{SS}	N7
AD16	M7	NC	A4	V_{CC}	J14	V_{SS}	N8
AD17	M6	NC	A5	V_{CC}	K1	V_{SS}	N9
AD18	P4	NC	B5	V_{CC}	K14	V_{SS}	N10
AD19	P3	NC	B14	V_{CC}	L14	V_{SS}	N11
AD20	N4	NC	C7	V_{CC}	P5	W/$\overline{R}$	B1
AD21	M5	NC	C8	V_{CC}	P6	WIDTH/HLTD0	B3
AD22	P2	NC	C14	V_{CC}	P7	WIDTH/HLTD1	A2
AD23	M4	NC	G12	V_{CC}	P8	$\overline{XINT0}$	C11
AD24	N3	NC	J12	V_{CC}	P9	$\overline{XINT1}$	C10
AD25	P1	NC	M3	V_{CC}	P10	$\overline{XINT2}$	A13
AD26	N2	$\overline{NMI}$	A10	V_{CC}	P11	$\overline{XINT3}$	B12
AD27	N1	$\overline{RDYRCV}$	F12	V_{CCPLL}	H12	$\overline{XINT4}$	B11
AD28	L3	$\overline{RESET}$	E12	V_{SS}	B6	$\overline{XINT5}$	A12
AD29	M2	STEST	C13	V_{SS}	B7	$\overline{XINT6}$	B10
AD30	M1	TCK	B13	V_{SS}	B8	$\overline{XINT7}$	A11

NOTE: Do not connect any external logic to pins marked NC (no connect pins).

Table 7. 132-Lead PGA Pinout — In Pin Order

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
A1	$\overline{ADS}$	C6	$\overline{FAIL}$	H1	V_{CC}	M10	AD12
A2	WIDTH/HLTD1	C7	NC	H2	V_{SS}	M11	AD9
A3	$\overline{ALE}$	C8	NC	H3	$\overline{BE0}$	M12	AD8
A4	NC	C9	HOLD	H12	V_{CCPLL}	M13	AD4
A5	NC	C10	$\overline{XINT1}$	H13	V_{SS}	M14	AD0
A6	V_{CC}	C11	$\overline{XINT0}$	H14	CLKIN	N1	AD27
A7	V_{CC}	C12	$\overline{TRST}$	J1	V_{CC}	N2	AD26
A8	V_{CC}	C13	STEST	J2	V_{SS}	N3	AD24
A9	V_{CC}	C14	NC	J3	$\overline{BE1}$	N4	AD20
A10	$\overline{NMI}$	D1	V_{CC}	J12	NC	N5	V_{SS}
A11	$\overline{XINT7}$	D2	V_{SS}	J13	V_{SS}	N6	V_{SS}
A12	$\overline{XINT5}$	D3	$\overline{DT/R}$	J14	V_{CC}	N7	V_{SS}
A13	$\overline{XINT2}$	D12	TDI	K1	V_{CC}	N8	V_{SS}
A14	TMS	D13	V_{SS}	K2	V_{SS}	N9	V_{SS}
B1	$W/\overline{R}$	D14	V_{CC}	K3	AD31	N10	V_{SS}
B2	$D/\overline{C}$	E1	V_{CC}	K12	AD2	N11	V_{SS}
B3	WIDTH/HLTD0	E2	V_{SS}	K13	V_{SS}	N12	AD10
B4	TDO	E3	$\overline{DEN}$	K14	V_{CC}	N13	AD7
B5	NC	E12	$\overline{RESET}$	L1	$\overline{BE2}$	N14	AD3
B6	V_{SS}	E13	V_{SS}	L2	$\overline{BE3}$	P1	AD25
B7	V_{SS}	E14	V_{CC}	L3	AD28	P2	AD22
B8	V_{SS}	F1	V_{CC}	L12	AD5	P3	AD19
B9	V_{SS}	F2	V_{SS}	L13	AD1	P4	AD18
B10	$\overline{XINT6}$	F3	BSTAT	L14	V_{CC}	P5	V_{CC}
B11	$\overline{XINT4}$	F12	$\overline{RDYRCV}$	M1	AD30	P6	V_{CC}
B12	$\overline{XINT3}$	F13	V_{SS}	M2	AD29	P7	V_{CC}
B13	TCK	F14	V_{CC}	M3	NC	P8	V_{CC}
B14	NC	G1	V_{CC}	M4	AD23	P9	V_{CC}
C1	$\overline{LOCK/ONCE}$	G2	V_{SS}	M5	AD21	P10	V_{CC}
C2	HOLDA	G3	ALE	M6	AD17	P11	V_{CC}
C3	$\overline{BLAST}$	G12	NC	M7	AD16	P12	AD13
C4	A3	G13	V_{SS}	M8	AD15	P13	AD11
C5	A2	G14	V_{CC}	M9	AD14	P14	AD6

NOTE: Do not connect any external logic to pins marked NC (no connect pins).

3.1.3 80960Jx PQFP Pinout

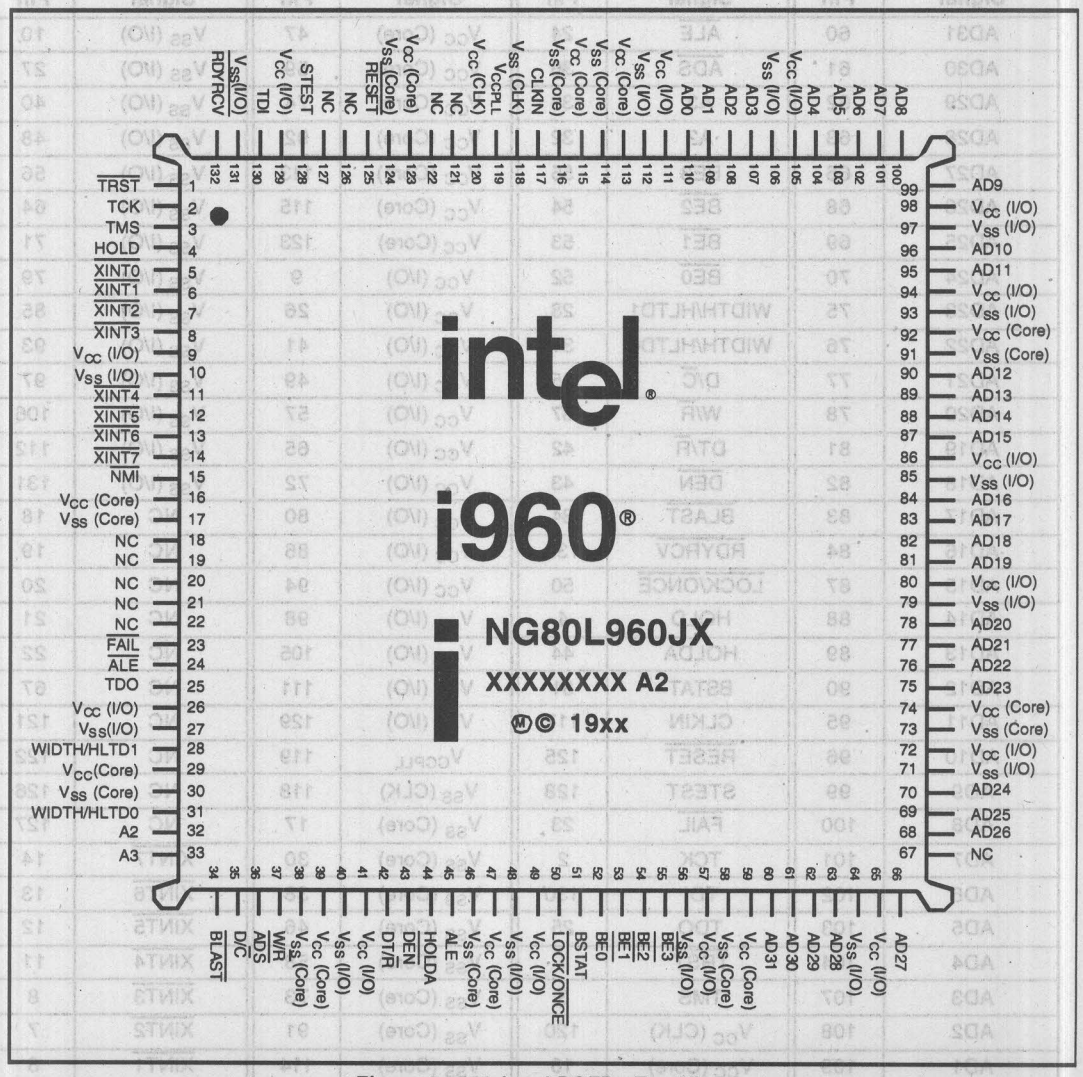


Figure 5. 132-Lead PQFP - Top View

Table 8. 132-Lead PQFP Pinout — In Signal Order

Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin
AD31	60	$\overline{ALE}$	24	V_{CC} (Core)	47	V_{SS} (I/O)	10
AD30	61	$\overline{ADS}$	36	V_{CC} (Core)	59	V_{SS} (I/O)	27
AD29	62	A3	33	V_{CC} (Core)	74	V_{SS} (I/O)	40
AD28	63	A2	32	V_{CC} (Core)	92	V_{SS} (I/O)	48
AD27	66	$\overline{BE3}$	55	V_{CC} (Core)	113	V_{SS} (I/O)	56
AD26	68	$\overline{BE2}$	54	V_{CC} (Core)	115	V_{SS} (I/O)	64
AD25	69	$\overline{BE1}$	53	V_{CC} (Core)	123	V_{SS} (I/O)	71
AD24	70	$\overline{BE0}$	52	V_{CC} (I/O)	9	V_{SS} (I/O)	79
AD23	75	WIDTH/HLTD1	28	V_{CC} (I/O)	26	V_{SS} (I/O)	85
AD22	76	WIDTH/HLTD0	31	V_{CC} (I/O)	41	V_{SS} (I/O)	93
AD21	77	D/$\overline{C}$	35	V_{CC} (I/O)	49	V_{SS} (I/O)	97
AD20	78	W/$\overline{R}$	37	V_{CC} (I/O)	57	V_{SS} (I/O)	106
AD19	81	DT/$\overline{R}$	42	V_{CC} (I/O)	65	V_{SS} (I/O)	112
AD18	82	$\overline{DEN}$	43	V_{CC} (I/O)	72	V_{SS} (I/O)	131
AD17	83	$\overline{BLAST}$	34	V_{CC} (I/O)	80	NC	18
AD16	84	$\overline{RDYRCV}$	132	V_{CC} (I/O)	86	NC	19
AD15	87	$\overline{LOCK/ONCE}$	50	V_{CC} (I/O)	94	NC	20
AD14	88	HOLD	4	V_{CC} (I/O)	98	NC	21
AD13	89	HOLDA	44	V_{CC} (I/O)	105	NC	22
AD12	90	BSTAT	51	V_{CC} (I/O)	111	NC	67
AD11	95	CLKIN	117	V_{CC} (I/O)	129	NC	121
AD10	96	$\overline{RESET}$	125	V_{CCPLL}	119	NC	122
AD9	99	STEST	128	V_{SS} (CLK)	118	NC	126
AD8	100	$\overline{FAIL}$	23	V_{SS} (Core)	17	NC	127
AD7	101	TCK	2	V_{SS} (Core)	30	$\overline{XINT7}$	14
AD6	102	TDI	130	V_{SS} (Core)	38	$\overline{XINT6}$	13
AD5	103	TDO	25	V_{SS} (Core)	46	$\overline{XINT5}$	12
AD4	104	$\overline{TRST}$	1	V_{SS} (Core)	58	$\overline{XINT4}$	11
AD3	107	TMS	3	V_{SS} (Core)	73	$\overline{XINT3}$	8
AD2	108	V_{CC} (CLK)	120	V_{SS} (Core)	91	$\overline{XINT2}$	7
AD1	109	V_{CC} (Core)	16	V_{SS} (Core)	114	$\overline{XINT1}$	6
AD0	110	V_{CC} (Core)	29	V_{SS} (Core)	116	$\overline{XINT0}$	5
ALE	45	V_{CC} (Core)	39	V_{SS} (Core)	124	$\overline{NMI}$	15

NOTE: Do not connect any external logic to pins marked NC (no connect pins).

5	$\overline{XINT0}$	38	V_{SS} (Core)	71	V_{SS} (I/O)	104	AD4
6	$\overline{XINT1}$	39	V_{CC} (Core)	72	V_{CC} (I/O)	105	V_{CC} (I/O)
7	$\overline{XINT2}$	40	V_{SS} (I/O)	73	V_{SS} (Core)	106	V_{SS} (I/O)
8	$\overline{XINT3}$	41	V_{CC} (I/O)	74	V_{CC} (Core)	107	AD3
9	V_{CC} (I/O)	42	DT/$\overline{R}$	75	AD23	108	AD2
10	V_{SS} (I/O)	43	$\overline{DEN}$	76	AD22	109	AD1
11	$\overline{XINT4}$	44	HOLDA	77	AD21	110	AD0
12	$\overline{XINT5}$	45	ALE	78	AD20	111	V_{CC} (I/O)
13	$\overline{XINT6}$	46	V_{SS} (Core)	79	V_{SS} (I/O)	112	V_{SS} (I/O)
14	$\overline{XINT7}$	47	V_{CC} (Core)	80	V_{CC} (I/O)	113	V_{CC} (Core)
15	$\overline{NMI}$	48	V_{SS} (I/O)	81	AD19	114	V_{SS} (Core)
16	V_{CC} (Core)	49	V_{CC} (I/O)	82	AD18	115	V_{CC} (Core)
17	V_{SS} (Core)	50	$\overline{LOCK/ONCE}$	83	AD17	116	V_{SS} (Core)
18	NC	51	BSTAT	84	AD16	117	CLKIN
19	NC	52	$\overline{BE0}$	85	V_{SS} (I/O)	118	V_{SS} (CLK)
20	NC	53	$\overline{BE1}$	86	V_{CC} (I/O)	119	V_{CCPLL}
21	NC	54	$\overline{BE2}$	87	AD15	120	$V_{CC\ (CLK)}$
22	NC	55	$\overline{BE3}$	88	AD14	121	NC
23	$\overline{FAIL}$	56	V_{SS} (I/O)	89	AD13	122	NC
24	$\overline{ALE}$	57	V_{CC} (I/O)	90	AD12	123	V_{CC} (Core)
25	TDO	58	V_{SS} (Core)	91	V_{SS} (Core)	124	V_{SS} (Core)
26	V_{CC} (I/O)	59	V_{CC} (Core)	92	V_{CC} (Core)	125	$\overline{RESET}$
27	V_{SS} (I/O)	60	AD31	93	V_{SS} (I/O)	126	NC
28	WIDTH/HLTD1	61	AD30	94	V_{CC} (I/O)	127	NC
29	V_{CC} (Core)	62	AD29	95	AD11	128	STEST
30	V_{SS} (Core)	63	AD28	96	AD10	129	V_{CC} (I/O)
31	WIDTH/HLTD0	64	V_{SS} (I/O)	97	V_{SS} (I/O)	130	TDI
32	A2	65	V_{CC} (I/O)	98	V_{CC} (I/O)	131	V_{SS} (I/O)
33	A3	66	AD27	99	AD9	132	$\overline{RDYRCV}$

NOTE: Do not connect any external logic to pins marked NC (no connect pins).

3.2 Package Thermal Specifications

The 80L960JA/JF is specified for operation when T_C (case temperature) is within the range of 0°C to 100°C. Case temperature may be measured in any environment to determine whether the 80L960JA/JF is within specified operating range. The case temperature should be measured at the center of the top surface, opposite the pins.

θ_{CA} is the thermal resistance from case to ambient. Use the following equation to calculate T_A, the maximum ambient temperature to conform to a particular case temperature:

$T_A = T_C - P\ (\theta_{CA})$

Junction temperature (T_J) is commonly used in reliability calculations. T_J can be calculated from θ_{JC} (thermal resistance from junction to case) using the following equation:

$T_J = T_C + P\ (\theta_{JC})$

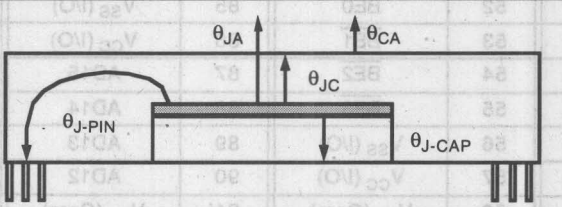
Similarly, if T_A is known, the corresponding case temperature (T_C) can be calculated as follows:

$T_C = T_A + P\ (\theta_{CA})$

Compute P by multiplying I_{CC} from Table 14 and V_{CC}. Values for θ_{JC} and θ_{CA} are given in Table 10 for the PGA package and Table 11 for the PQFP package. For high speed operation, the processor's θ_{JA} may be significantly reduced by adding a heatsink and/or by increasing airflow.

Table 10. 132-Lead PGA Package Thermal Characteristics

Thermal Resistance — °C/Watt						
Parameter	Airflow — ft./min (m/sec)					
	0 (0)	200 (1.01)	400 (2.03)	600 (3.04)	800 (4.06)	1000 (5.08)
θ_{JC} (Junction-to-Case)	3	3	3	3	3	3
θ_{CA} (Case-to-Ambient) (No Heatsink)	18	15	12	11	11	11
θ_{CA} (Case-to-Ambient) (Omnidirectional Heatsink)	15	12	9	8	8	8
θ_{CA} (Case-to-Ambient) (Unidirectional Heatsink)	14	11	8	7	7	7

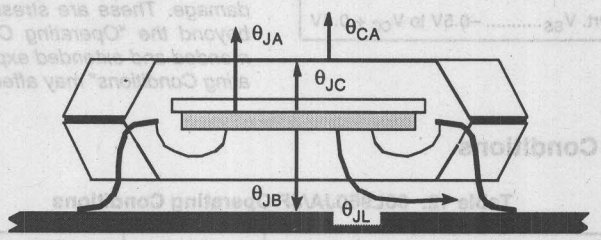


NOTES:

1. This table applies to a PGA device plugged into a socket or soldered directly into a board.
2. $\theta_{JA} = \theta_{JC} + \theta_{CA}$
3. $\theta_{J\text{-}CAP} = 4°C/W$ (approx.)
4. $\theta_{J\text{-}PIN} = 4°C/W$ (inner pins) (approx.)
5. $\theta_{J\text{-}PIN} = 8°C/W$ (outer pins) (approx.)

Table 11. 132-Lead PQFP Package Thermal Characteristics

Thermal Resistance — °C/Watt							
Parameter	**Airflow — ft./min (m/sec)**						
	0 (0)	**50 (0.25)**	**100 (0.50)**	**200 (1.01)**	**400 (2.03)**	**600 (3.04)**	**800 (4.06)**
θ_{JC} (Junction-to-Case)	6	7	7	7	7	7	7
θ_{CA} (Case-to-Ambient -No Heatsink)	23	20	18	14	10	9	8



NOTES:

1. This table applies to a PQFP device soldered directly into board.
2. $\theta_{JA} = \theta_{JC} + \theta_{CA}$
3. θ_{JL} = 18°C/W (approx.)
4. θ_{JB} = 18°C/W (approx.)

4.0 ELECTRICAL SPECIFICATIONS

4.1 Absolute Maximum Ratings

Parameter	Maximum Rating
Storage Temperature	−65° C to +150° C
Case Temperature Under Bias	−65° C to +110° C
Supply Voltage wrt. V_{SS}	−0.5V to + 4.6V
Voltage on Other Pins wrt. V_{SS}	−0.5V to V_{CC} + 0.5V

NOTICE: This data sheet contains preliminary information on new products in production. The specifications are subject to change without notice.

WARNING: *Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

4.2 Operating Conditions

Table 12. 80L960JA/JF Operating Conditions

Symbol	Parameter	Min	Max	Units	Notes
V_{CC}	Supply Voltage				
	80L960JA/JF-25	3.0	3.6	V	
	80L960JA/JF-16	3.0	3.6		
f_{CLKIN}	Input Clock Frequency				
	80L960JA/JF-25	8	25	MHz	
	80L960JA/JF-16	8	16.67		
T_C	Operating Case Temperature				
	A80L960JA/JF-25 (132 PGA)	0	100	°C	
	A80L960JA/JF-16 (132 PGA)	0	100		
	NG80L960JA/JF-25 (132 PQFP)	0	100		
	NG80L960JA/JF-16 (132 PQFP)	0	100		

80L960JA/JF, since the processor can cause transient power surges.

pins if the TAP Controller will not be used.

Pins identified as NC must not be connected in the system.

4.4 DC Specifications

Table 13. 80L960JA/JF DC Characteristics

Symbol	Parameter	Min	Typ	Max	Units	Notes
V_{IL}	Input Low Voltage	-0.3		0.8	V	
V_{IH}	Input High Voltage	2.0		V_{CC} + 0.3	V	
V_{OL}	Output Low Voltage			0.4 0.2		I_{OL} = 3 mA I_{OL} = 100 µA
V_{OH}	Output High Voltage	2.4 V_{CC} - 0.2		V	V	I_{OH} = -3 mA I_{OH} = -100 µA
V_{OLP}	Output Ground Bounce		< 0.8		V	(1,2)
C_{IN}	Input Capacitance PGA PQFP			 12 10	 pF	$f_{CLKIN} = f_{MIN}$ (2)
C_{OUT}	I/O or Output Capacitance PGA PQFP			 12 10	 pF	$f_{CLKIN} = f_{MIN}$ (2)
C_{CLK}	CLKIN Capacitance PGA PQFP			 12 10	 pF	$f_{CLKIN} = f_{MIN}$ (2)

NOTES:

1. Typical is measured with V_{CC} = 3.3V and temperature = 25 °C.
2. Not tested.

Table 14. 80L960JA/JF I_{CC} Characteristics

Symbol	Parameter	Typ	Max	Units	Notes
I_{LI1}	Input Leakage Current for each pin except TCK, TDI, TRST and TMS		± 1	µA	$0 \le V_{IN} \le V_{CC}$
I_{LI2}	Input Leakage Current for TCK, TDI, TRST and TMS	-140	-250	µA	V_{IN} = 0.45V (1)
I_{LO}	Output Leakage Current		± 1	µA	$0.4 \le V_{OUT} \le V_{CC}$
I_{CC} Active (Power Supply)	80960JA/JF-25		284	mA	(2,3)
	80960JA/JF-16		184		(2,3)
I_{CC} Active (Thermal)	80960JA/JF-25	225		mA	(2,4)
	80960JA/JF-16	165			(2,4)
I_{CC} Test (Power modes)	Reset mode				
	80960JA/JF-25		200	mA	(5)
	80960JA/JF-16		150		(5)
	Halt mode				
	80960JA/JF-25		26		(5)
	80960JA/JF-16		16		(5)
	ONCE mode		10		(5)

NOTES:

1. These pins have internal pullup devices. Typical leakage current is not tested.
2. Measured with device operating and outputs loaded to the test condition in Figure 6, AC Test Load (pg. 30).
3. I_{CC} Active (Power Supply) value is provided for selecting your system's power supply. It is measured using one of the worst case instruction mixes with V_{CC} = 3.6V. This parameter is characterized but not tested.
4. I_{CC} Active (Thermal) value is provided for your system's thermal management. Typical I_{CC} is measured with V_{CC} = 3.3V and temperature = 25° C. This parameter is characterized but not tested.
5. I_{CC} Test (Power modes) refers to the I_{CC} values that are tested when the processor is in Reset mode, Halt mode or ONCE mode with V_{CC} = 3.6V.

4.5 AC Specifications

The 80L960JA/JF AC timings are based upon device characterization.

Table 15. 80L960JA/JF AC Characteristics (25 MHz) (Sheet 1 of 2)

Symbol	Parameter	Min	Max	Units	Notes
	INPUT CLOCK TIMINGS				
T_F	CLKIN Frequency	8	25	MHz	
T_C	CLKIN Period	40	125	ns	
T_{CS}	CLKIN Period Stability		± 250	ps	(1,2)
T_{CH}	CLKIN High Time	12		ns	Measured at 1.5 V (1)
T_{CL}	CLKIN Low Time	12		ns	Measured at 1.5 V (1)
T_{CR}	CLKIN Rise Time		4	ns	0.8 V to 2.0 V (1)
T_{CF}	CLKIN Fall Time		4	ns	2.0 V to 0.8 V (1)
	SYNCHRONOUS OUTPUT TIMINGS				
T_{OV1}	Output Valid Delay, Except ALE/$\overline{ALE}$ Inactive and DT/$\overline{R}$	4.0	18	ns	(3)
T_{OV2}	Output Valid Delay, DT/$\overline{R}$	$0.5T_C + 4.0$	$0.5T_C + 18$	ns	
T_{OF}	Output Float Delay	4.0	16	ns	(4)
	SYNCHRONOUS INPUT TIMINGS				
T_{IS1}	Input Setup to CLKIN — AD31:0, $\overline{NMI}$, $\overline{XINT7:0}$	9		ns	(5)
T_{IH1}	Input Hold from CLKIN — AD31:0, $\overline{NMI}$, $\overline{XINT7:0}$	1		ns	(5)
T_{IS2}	Input Setup to CLKIN — $\overline{RDYRCV}$ and HOLD	10		ns	(6)
T_{IH2}	Input Hold from CLKIN — $\overline{RDYRCV}$ and HOLD	1		ns	(6)
T_{IS3}	Input Setup to CLKIN — $\overline{RESET}$	8		ns	(7)
T_{IH3}	Input Hold from CLKIN — $\overline{RESET}$	1		ns	(7)
T_{IS4}	Input Setup to $\overline{RESET}$ — $\overline{ONCE}$, STEST	8		ns	(8)

NOTE: See Table 16 on page 27 for note definitions for this table.

Table 15. 80L960JA/JF AC Characteristics (25 MHz) (Sheet 2 of 2)

Symbol	Parameter	Min	Max	Units	Notes
T_{IH4}	Input Hold from $\overline{RESET}$ — $\overline{ONCE}$, STEST	1		ns	(8)
RELATIVE OUTPUT TIMINGS					
T_{LXL}	ALE/$\overline{ALE}$ Width	$0.5T_C$ - 7.5		ns	(9)
T_{LXA}	Address Hold from ALE/$\overline{ALE}$ Inactive				Equal Loading (9)
T_{DXD}	DT/$\overline{R}$ Valid to $\overline{DEN}$ Active				Equal Loading (9)
BOUNDARY SCAN TEST SIGNAL TIMINGS					
T_{BSF}	TCK Frequency		$0.5T_F$	MHz	
T_{BSCH}	TCK High Time	15		ns	Measured at 1.5 V (1)
T_{BSCL}	TCK Low Time	15		ns	Measured at 1.5 V (1)
T_{BSCR}	TCK Rise Time		5	ns	0.8 V to 2.0 V (1)
T_{BSCF}	TCK Fall Time		5	ns	2.0 V to 0.8 V (1)
T_{BSIS1}	Input Setup to TCK — TDI, TMS	4		ns	
T_{BSIH1}	Input Hold from TCK — TDI, TMS	6		ns	
T_{BSOV1}	TDO Valid Delay	3	30	ns	(1, 10)
T_{BSOF1}	TDO Float Delay	3	36	ns	(1, 10)
T_{BSOV2}	All Outputs (Non-Test) Valid Delay	3	35	ns	(1, 10)
T_{BSOF2}	All Outputs (Non-Test) Float Delay	3	36	ns	(1, 10)
T_{BSIS2}	Input Setup to TCK — All Inputs (Non-Test)	4		ns	
T_{BSIH2}	Input Hold from TCK — All Inputs (Non-Test)	6		ns	

NOTE: See Table 16 on page 27 for note definitions for this table.

Table 16. Note Definitions for Table 15, 80L960JA/JF AC Characteristics (25 MHz) (pg. 25)

NOTES:

1. Not tested.
2. To ensure a 1:1 relationship between the amplitude of the input jitter and the internal clock, the jitter frequency spectrum should not have any power peaking between 500 KHz and 1/3 of the CLKIN frequency.
3. Inactive ALE/$\overline{ALE}$ refers to the falling edge of ALE and the rising edge of $\overline{ALE}$. For inactive ALE/$\overline{ALE}$ timings, refer to Relative Output Timings in this table.
4. A float condition occurs when the output current becomes less than I_{LO}. Float delay is not tested, but is designed to be no longer than the valid delay.
5. AD31:0 are synchronous inputs. Setup and hold times must be met for proper processor operation. $\overline{NMI}$ and XINT7:0 may be synchronous or asynchronous. Meeting setup and hold time guarantees recognition at a particular clock edge. For asynchronous operation, $\overline{NMI}$ and XINT7:0 must be asserted for a minimum of two CLKIN periods to guarantee recognition.
6. $\overline{RDYRCV}$ and HOLD are synchronous inputs. Setup and hold times must be met for proper processor operation.
7. $\overline{RESET}$ may be synchronous or asynchronous. Meeting setup and hold time guarantees recognition at a particular clock edge.
8. $\overline{ONCE}$ and $\overline{STEST}$ must be stable at the rising edge of $\overline{RESET}$ for proper operation.
9. Guaranteed by design. May not be 100% tested.
10. Relative to falling edge of TCK.

Table 17. 80L960JA/JF AC Characteristics (16 MHz) (Sheet 1 of 3)

Symbol	Parameter	Min	Max	Units	Notes
	INPUT CLOCK TIMINGS				
T_F	CLKIN Frequency	8	16.67	MHz	
T_C	CLKIN Period	60	125	ns	
T_{CS}	CLKIN Period Stability		± 250	ps	(1,2)
T_{CH}	CLKIN High Time	18		ns	Measured at 1.5 V (1)
T_{CL}	CLKIN Low Time	18		ns	Measured at 1.5 V (1)
T_{CR}	CLKIN Rise Time		6	ns	0.8 V to 2.0 V (1)
T_{CF}	CLKIN Fall Time		6	ns	2.0 V to 0.8 V (1)
	SYNCHRONOUS OUTPUT TIMINGS				
T_{OV1}	Output Valid Delay, Except ALE/$\overline{ALE}$ Inactive and DT/$\overline{R}$	4.0	21	ns	(3)
T_{OV2}	Output Valid Delay, DT/$\overline{R}$	$0.5T_C + 4.0$	$0.5T_C + 21$	ns	
T_{OF}	Output Float Delay	4.0	19	ns	(4)

Table 17. 80L960JA/JF AC Characteristics (16 MHz) (Sheet 2 of 3)

Symbol	Parameter	Min	Max	Units	Notes
	SYNCHRONOUS INPUT TIMINGS				
T_{IS1}	Input Setup to CLKIN — AD31:0, $\overline{NMI}$, $\overline{XINT7:0}$	10		ns	(5)
T_{IH1}	Input Hold from CLKIN — AD31:0, $\overline{NMI}$, $\overline{XINT7:0}$	1		ns	(5)
T_{IS2}	Input Setup to CLKIN — $\overline{RDYRCV}$ and HOLD	11		ns	(6)
T_{IH2}	Input Hold from CLKIN — $\overline{RDYRCV}$ and HOLD	1		ns	(6)
T_{IS3}	Input Setup to CLKIN — $\overline{RESET}$	8		ns	(7)
T_{IH3}	Input Hold from CLKIN — $\overline{RESET}$	1		ns	(7)
T_{IS4}	Input Setup to $\overline{RESET}$ — $\overline{ONCE}$, STEST	8		ns	(8)
T_{IH4}	Input Hold from $\overline{RESET}$ — $\overline{ONCE}$, STEST	1		ns	(8)
	RELATIVE OUTPUT TIMINGS				
T_{LXL}	ALE/$\overline{ALE}$ Width	$0.5T_C$ - 8		ns	(9)
T_{LXA}	Address Hold from ALE/$\overline{ALE}$ Inactive				Equal Loading (9)
T_{DXD}	DT/$\overline{R}$ Valid to $\overline{DEN}$ Active				Equal Loading (9)
	BOUNDARY SCAN TEST SIGNAL TIMINGS				
T_{BSF}	TCK Frequency		$0.5T_F$	MHz	
T_{BSCH}	TCK High Time	15		ns	Measured at 1.5 V (1)
T_{BSCL}	TCK Low Time	15		ns	Measured at 1.5 V (1)
T_{BSCR}	TCK Rise Time		5	ns	0.8 V to 2.0 V (1)
T_{BSCF}	TCK Fall Time		5	ns	2.0 V to 0.8 V (1)
T_{BSIS1}	Input Setup to TCK — TDI, TMS	4		ns	
T_{BSIH1}	Input Hold from TCK — TDI, TMS	6		ns	
T_{BSOV1}	TDO Valid Delay	3	30	ns	(1, 10)
T_{BSOF1}	TDO Float Delay	3	36	ns	(1, 10)

Table 17. 80L960JA/JF AC Characteristics (16 MHz) (Sheet 3 of 3)

Symbol	Parameter	Min	Max	Units	Notes
T_{BSQV2}	All Outputs (Non-Test) Valid Delay	3	35	ns	(1, 10)
T_{BSOF2}	All Outputs (Non-Test) Float Delay	3	36	ns	(1, 10)
T_{BSIS2}	Input Setup to TCK — All Inputs (Non-Test)	4		ns	
T_{BSIH2}	Input Hold from TCK — All Inputs (Non-Test)	6		ns	

NOTES:

1. Not tested.
2. To ensure a 1:1 relationship between the amplitude of the input jitter and the internal clock, the jitter frequency spectrum should not have any power peaking between 500 KHz and 1/3 of the CLKIN frequency.
3. Inactive ALE/$\overline{\text{ALE}}$ refers to the falling edge of ALE and the rising edge of $\overline{\text{ALE}}$. For inactive ALE/$\overline{\text{ALE}}$ timings, refer to Relative Output Timings in this table.
4. A float condition occurs when the output current becomes less than I_{LO}. Float delay is not tested, but is designed to be no longer than the valid delay.
5. AD31:0 are synchronous inputs. Setup and hold times must be met for proper processor operation. $\overline{\text{NMI}}$ and $\overline{\text{XINT7:0}}$ may be synchronous or asynchronous. Meeting setup and hold time guarantees recognition at a particular clock edge. For asynchronous operation, $\overline{\text{NMI}}$ and $\overline{\text{XINT7:0}}$ must be asserted for a minimum of two CLKIN periods to guarantee recognition.
6. $\overline{\text{RDYRCV}}$ and HOLD are synchronous inputs. Setup and hold times must be met for proper processor operation.
7. $\overline{\text{RESET}}$ may be synchronous or asynchronous. Meeting setup and hold time guarantees recognition at a particular clock edge.
8. $\overline{\text{ONCE}}$ and $\overline{\text{STEST}}$ must be stable at the rising edge of $\overline{\text{RESET}}$ for proper operation.
9. Guaranteed by design. May not be 100% tested.
10. Relative to falling edge of TCK.

4.5.1 AC Test Conditions and Derating Curves

The AC Specifications in **Section 4.5, AC Specifications** are tested with the 50 pF load indicated in Figure 6. Figure 7 shows how timings vary with load capacitance; Figure 8 shows how output rise and fall times vary with load capacitance.

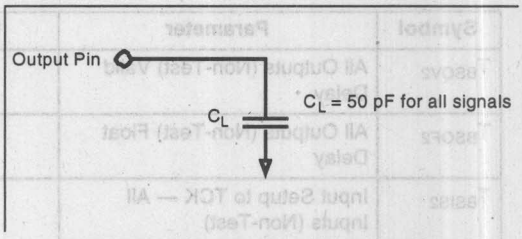


Figure 6. AC Test Load

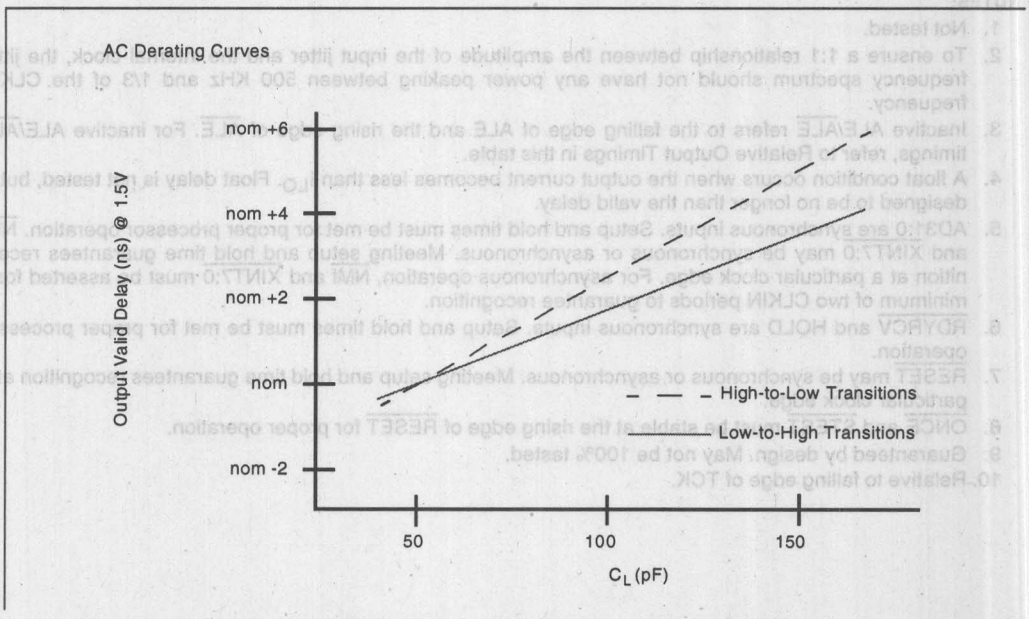


Figure 7. Output Delay or Hold vs. Load Capacitance

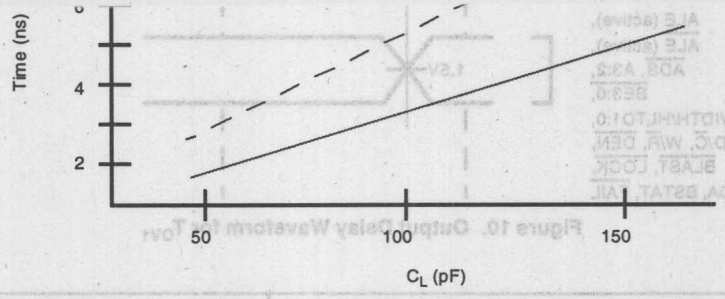


Figure 8. Rise and Fall Time Derating

4.5.2 AC Timing Waveforms

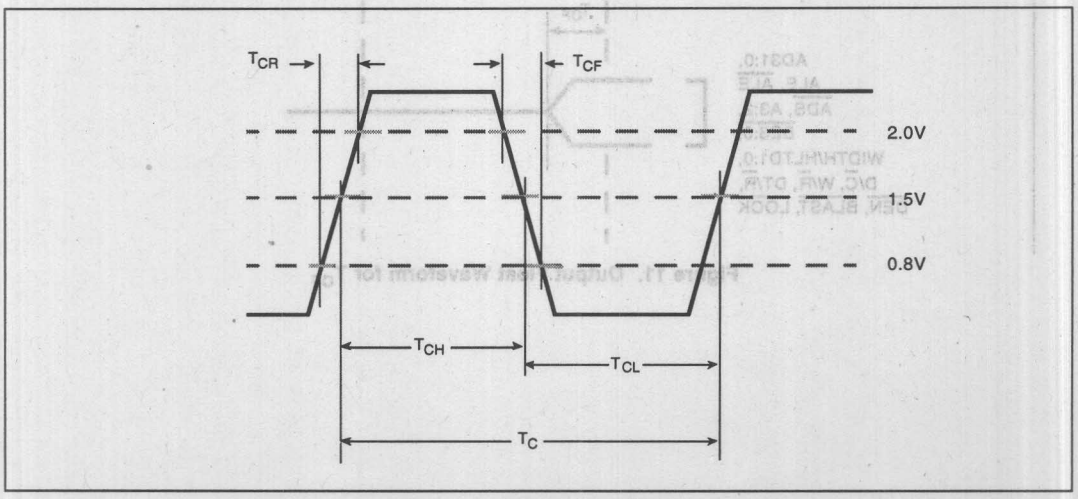


Figure 9. CLKIN Waveform

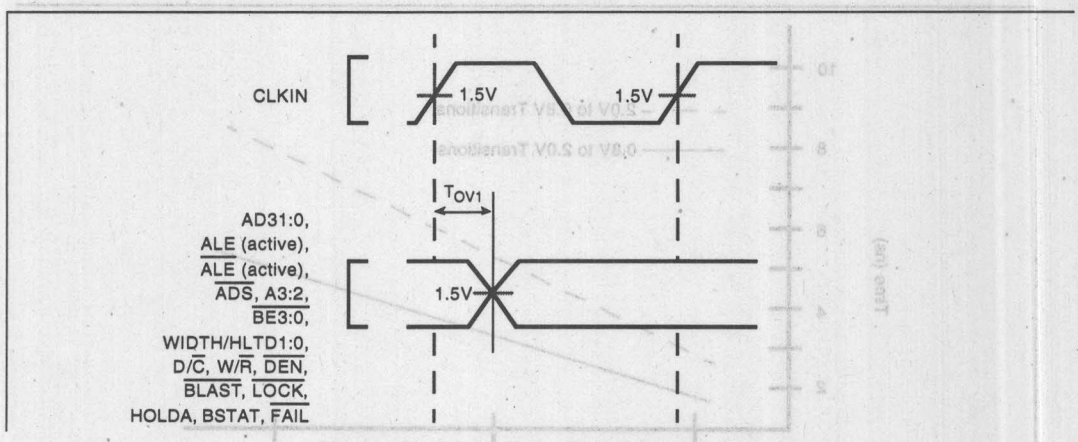


Figure 10. Output Delay Waveform for T_{OV1}

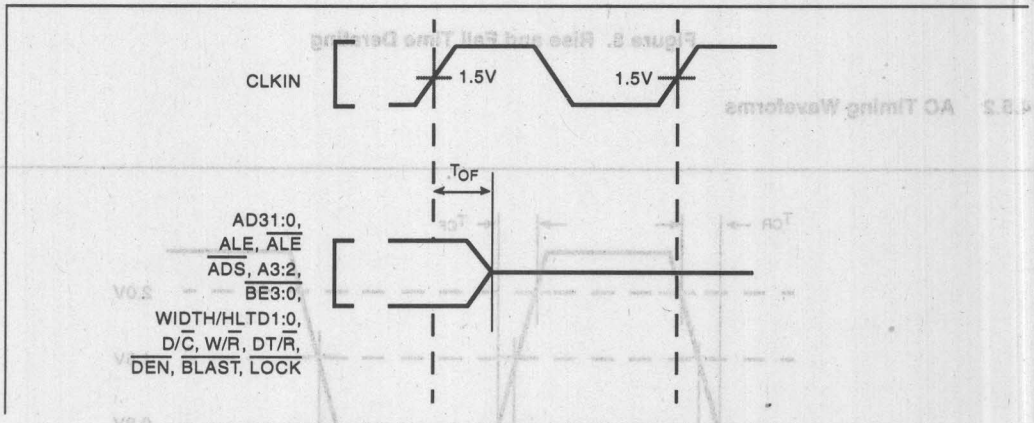


Figure 11. Output Float Waveform for T_{OF}

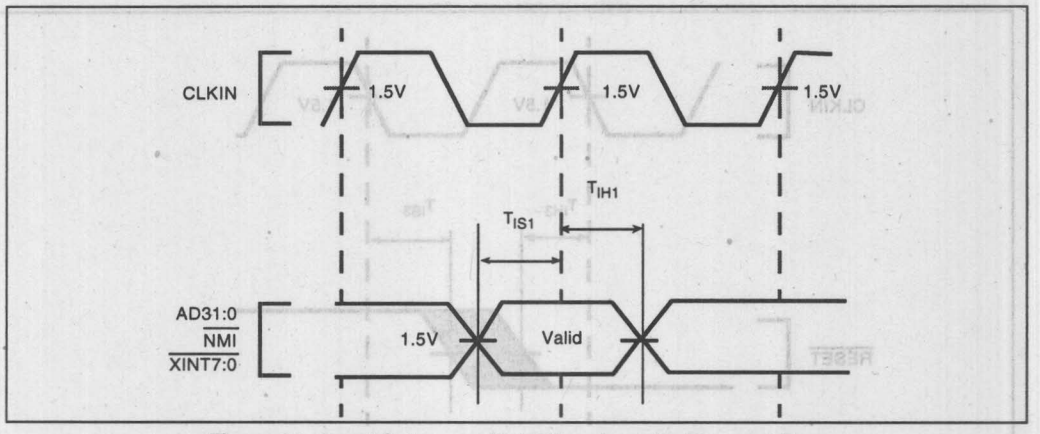


Figure 12. Input Setup and Hold Waveform for T_{IS1} and T_{IH1}

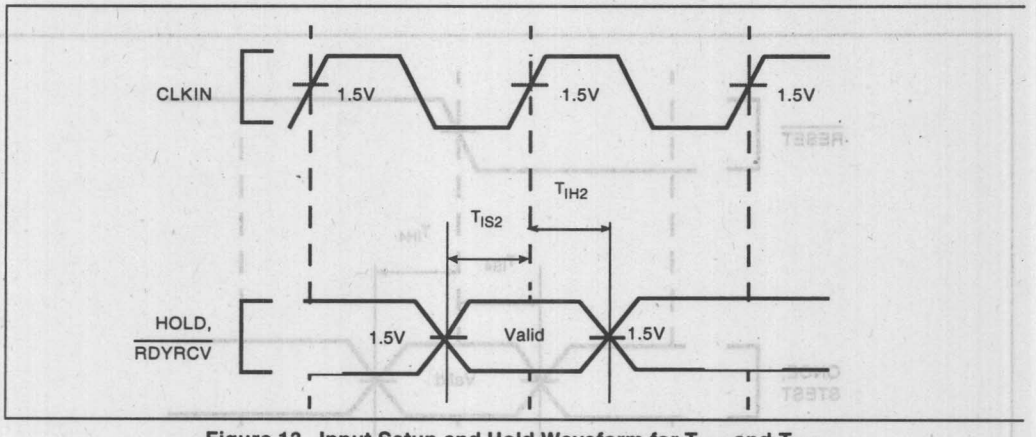


Figure 13. Input Setup and Hold Waveform for T_{IS2} and T_{IH2}

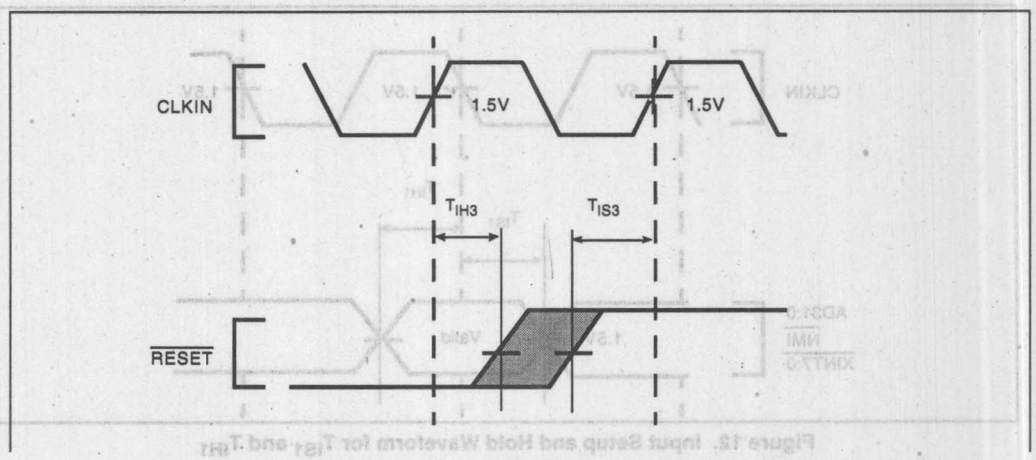


Figure 14. Input Setup and Hold Waveform for T_{IS3} and T_{IH3}

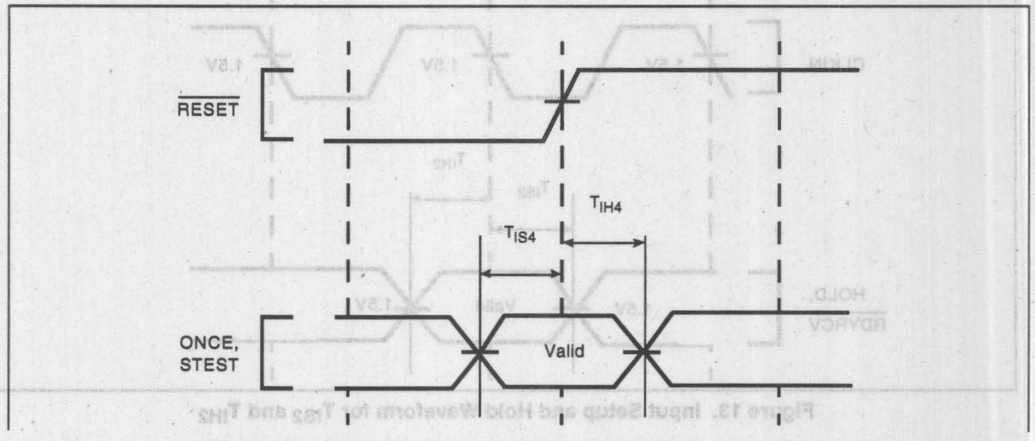


Figure 15. Input Setup and Hold Waveform for T_{IS4} and T_{IH4}

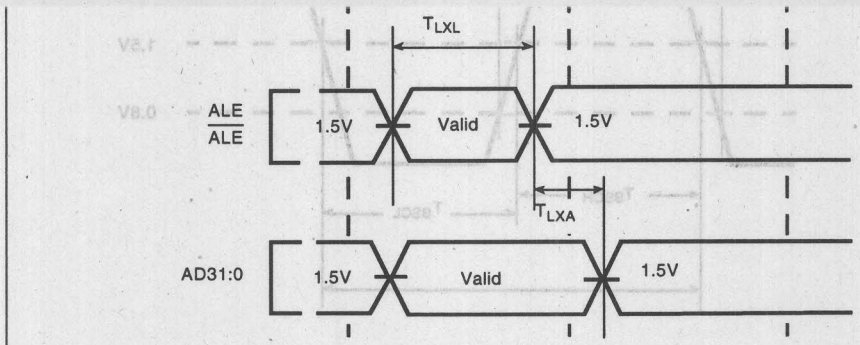


Figure 16. Relative Timings Waveform for T_{LXL} and T_{LXA}

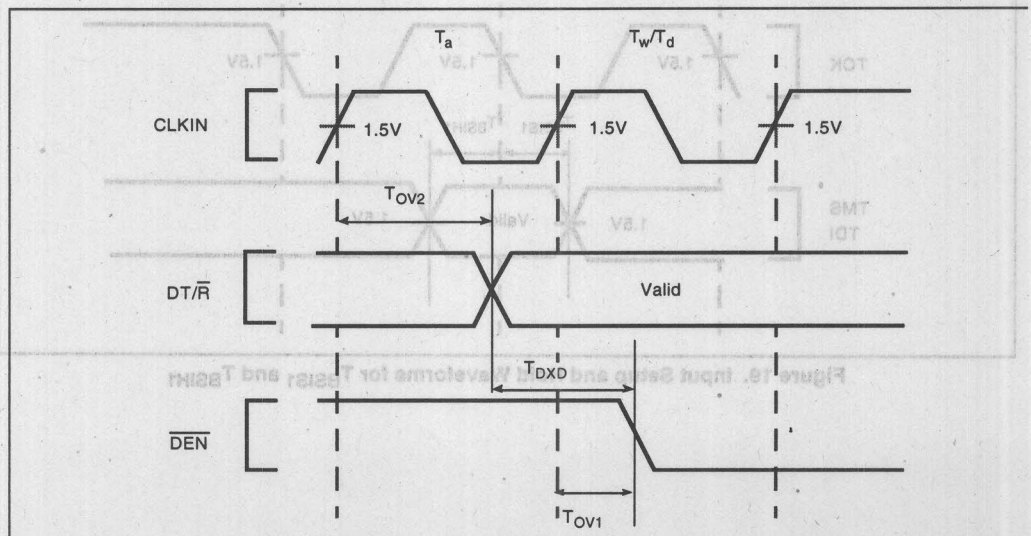


Figure 17. DT/$\overline{R}$ and $\overline{DEN}$ Timings Waveform

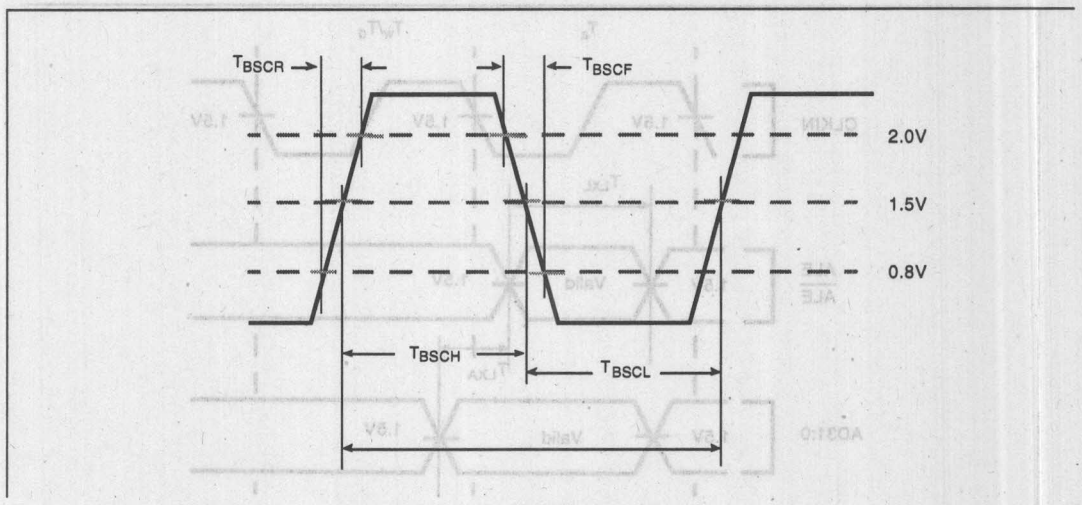


Figure 18. TCK Waveform

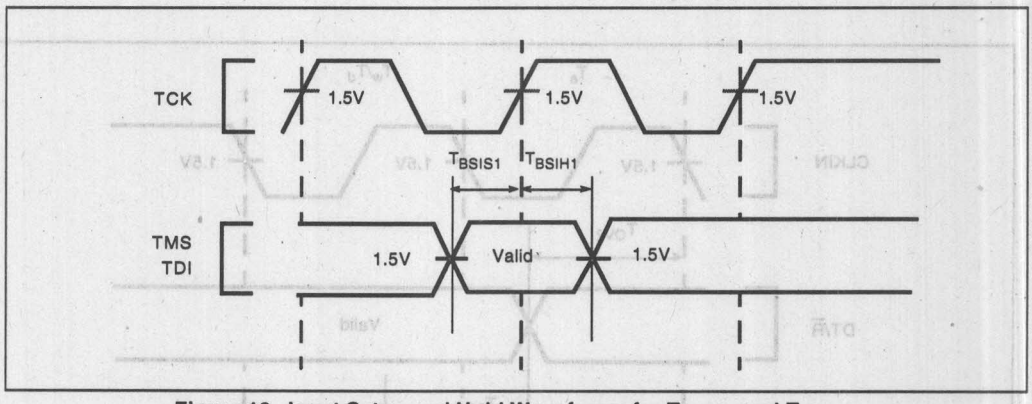


Figure 19. Input Setup and Hold Waveforms for T_{BSIS1} and T_{BSIH1}

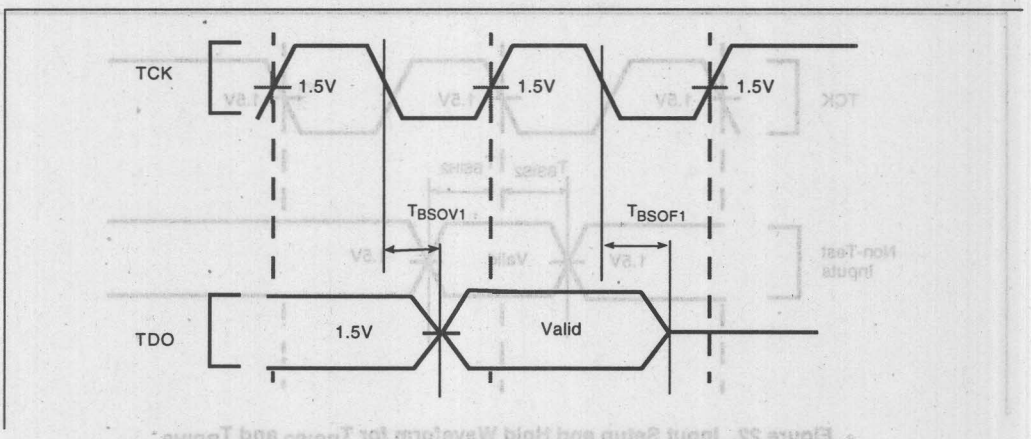


Figure 20. Output Delay and Output Float Waveform for T_{BSOV1} AND T_{BSOF1}

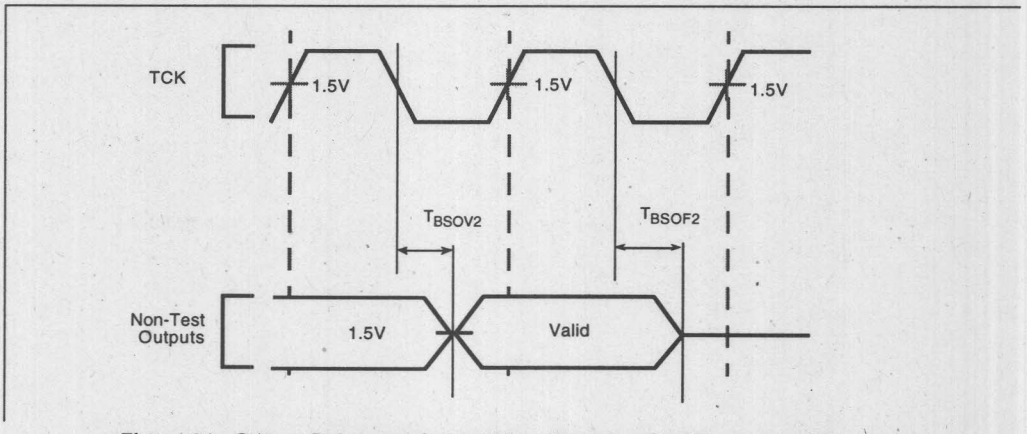


Figure 21. Output Delay and Output Float Waveform for T_{BSOV2} and T_{BSOF2}

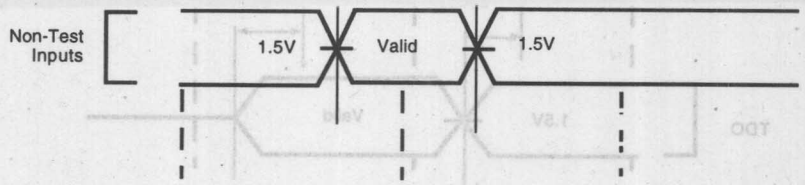


Figure 22. Input Setup and Hold Waveform for T_{BSIS2} and T_{BSIH2}

5.0 BUS FUNCTIONAL WAVEFORMS

Figures 23 through 28 illustrate typical 80L960JA/JF bus transactions. Figure 29 depicts the bus arbitration sequence. Figure 30 illustrates the processor reset sequence from the time power is applied to the device. Figure 31 illustrates the processor reset sequence when the processor is in operation. Figure 32 illustrates the processor ONCE sequence from the time power is applied to the device. Figures 33 and 34 also show accesses on 32-bit buses. Tables 18 through 21 summarize all possible combinations of bus accesses across 8-, 16-, and 32-bit buses according to data alignment.

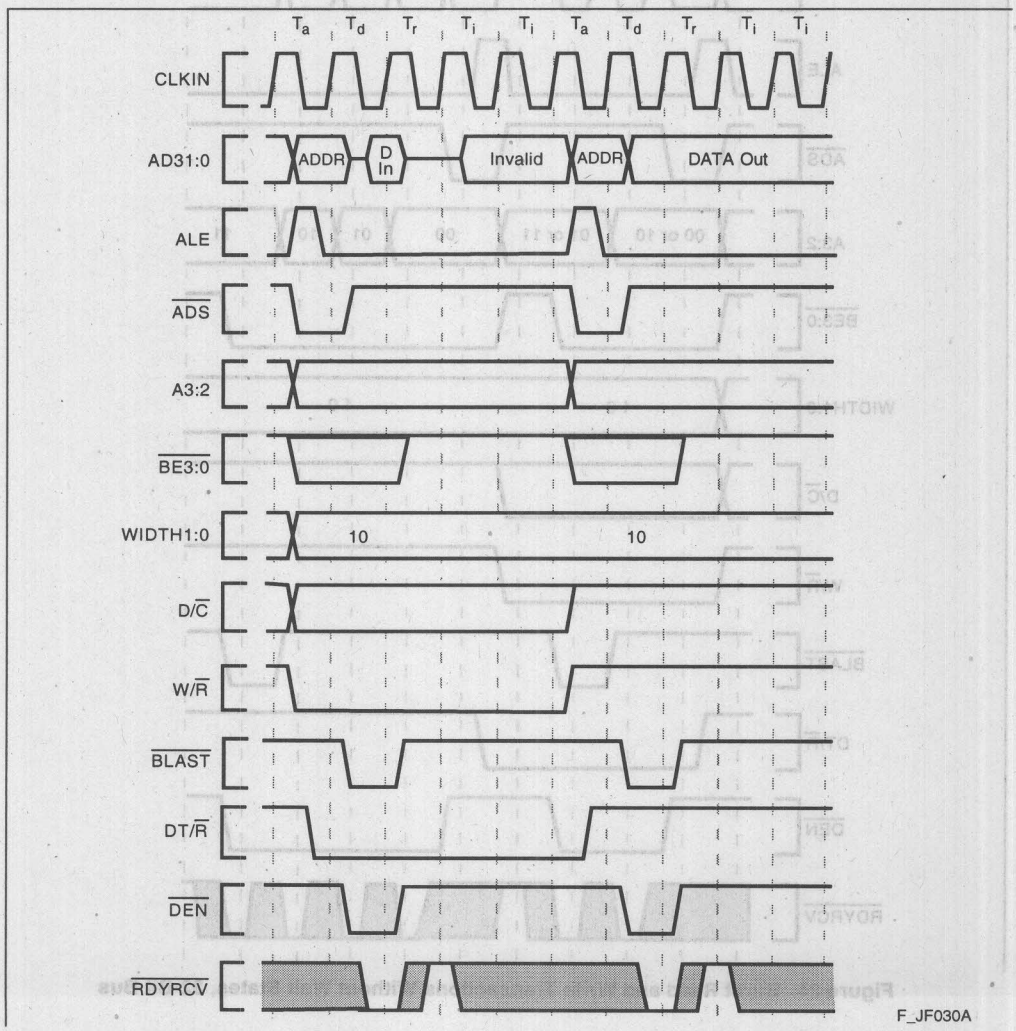


Figure 23. Non-Burst Read and Write Transactions Without Wait States, 32-Bit Bus

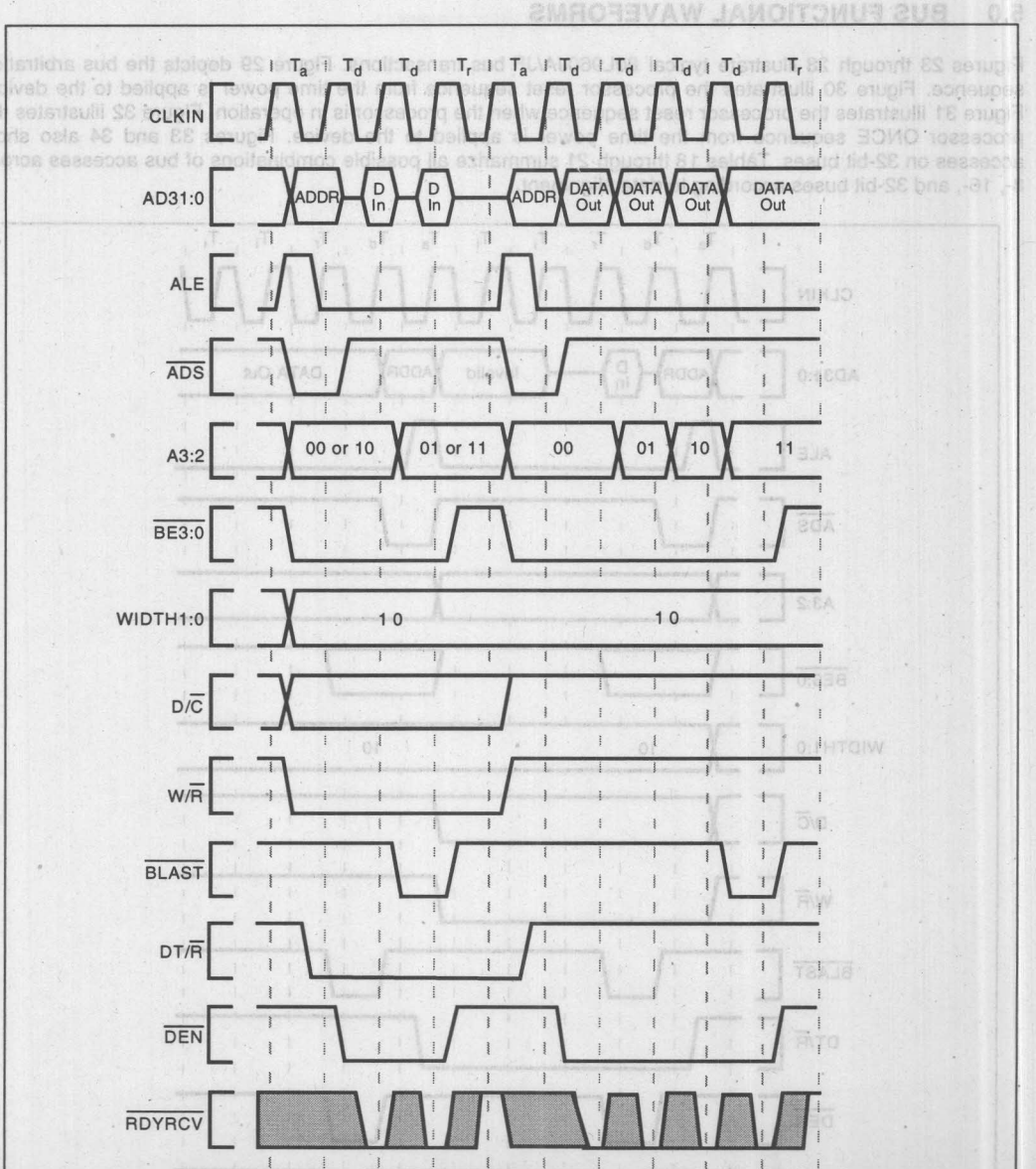


Figure 24. Burst Read and Write Transactions Without Wait States, 32-Bit Bus

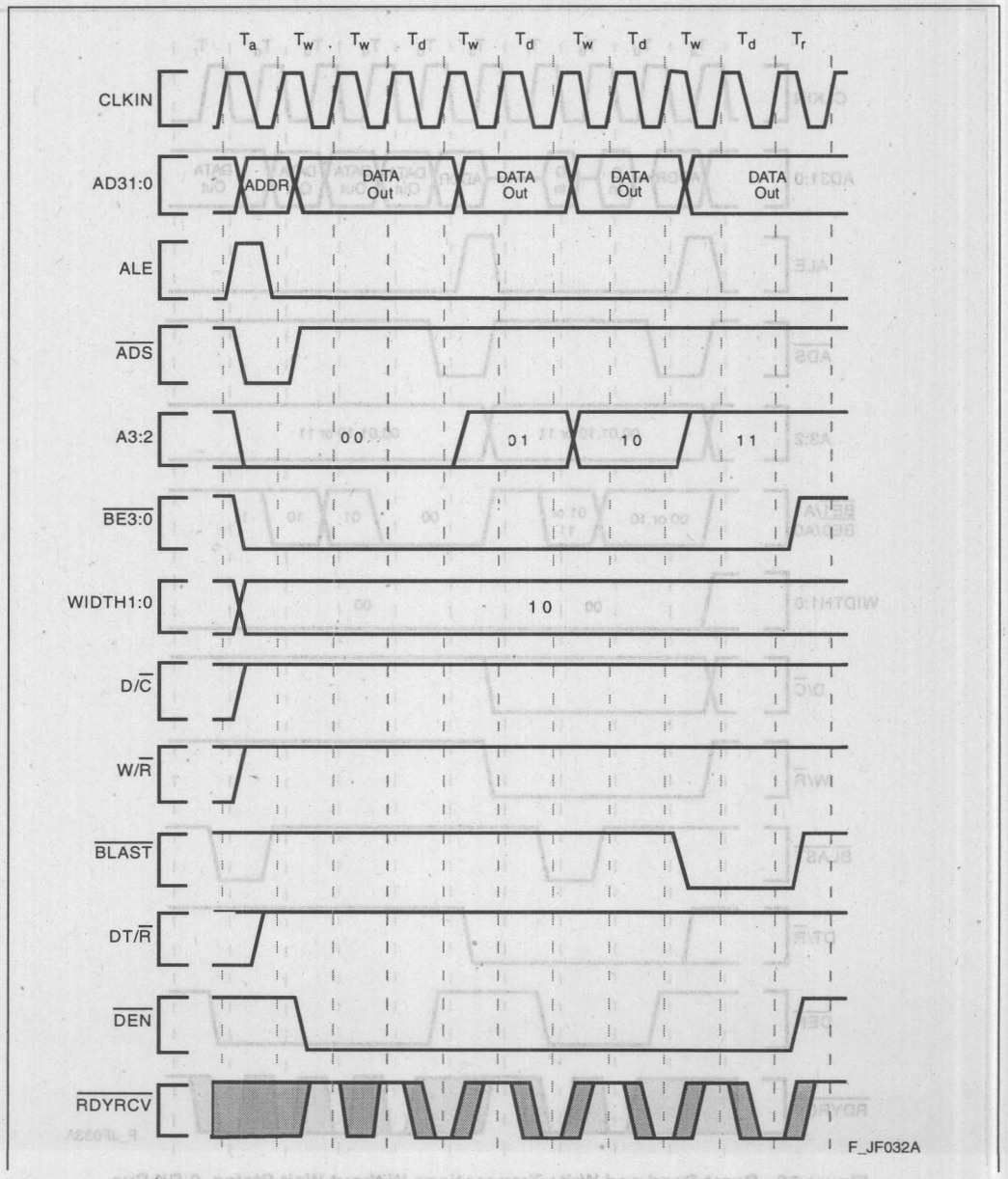


Figure 25. Burst Write Transactions With 2,1,1,1 Wait States, 32-Bit Bus

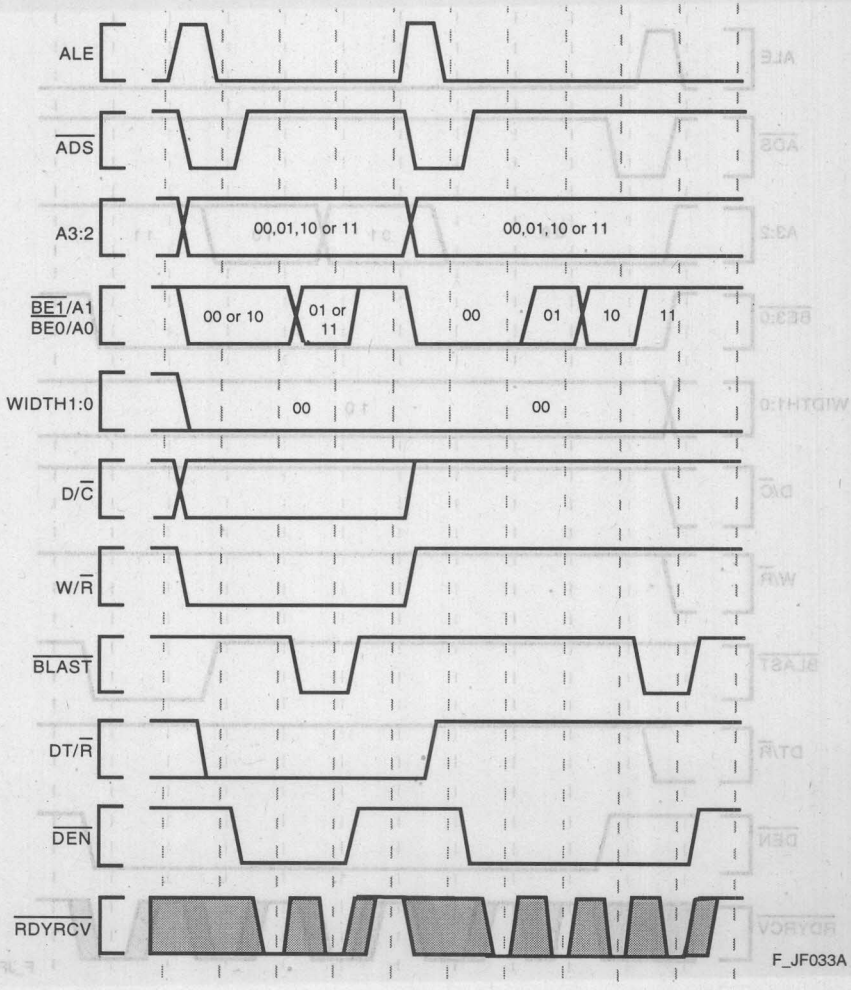


Figure 26. Burst Read and Write Transactions Without Wait States, 8-Bit Bus

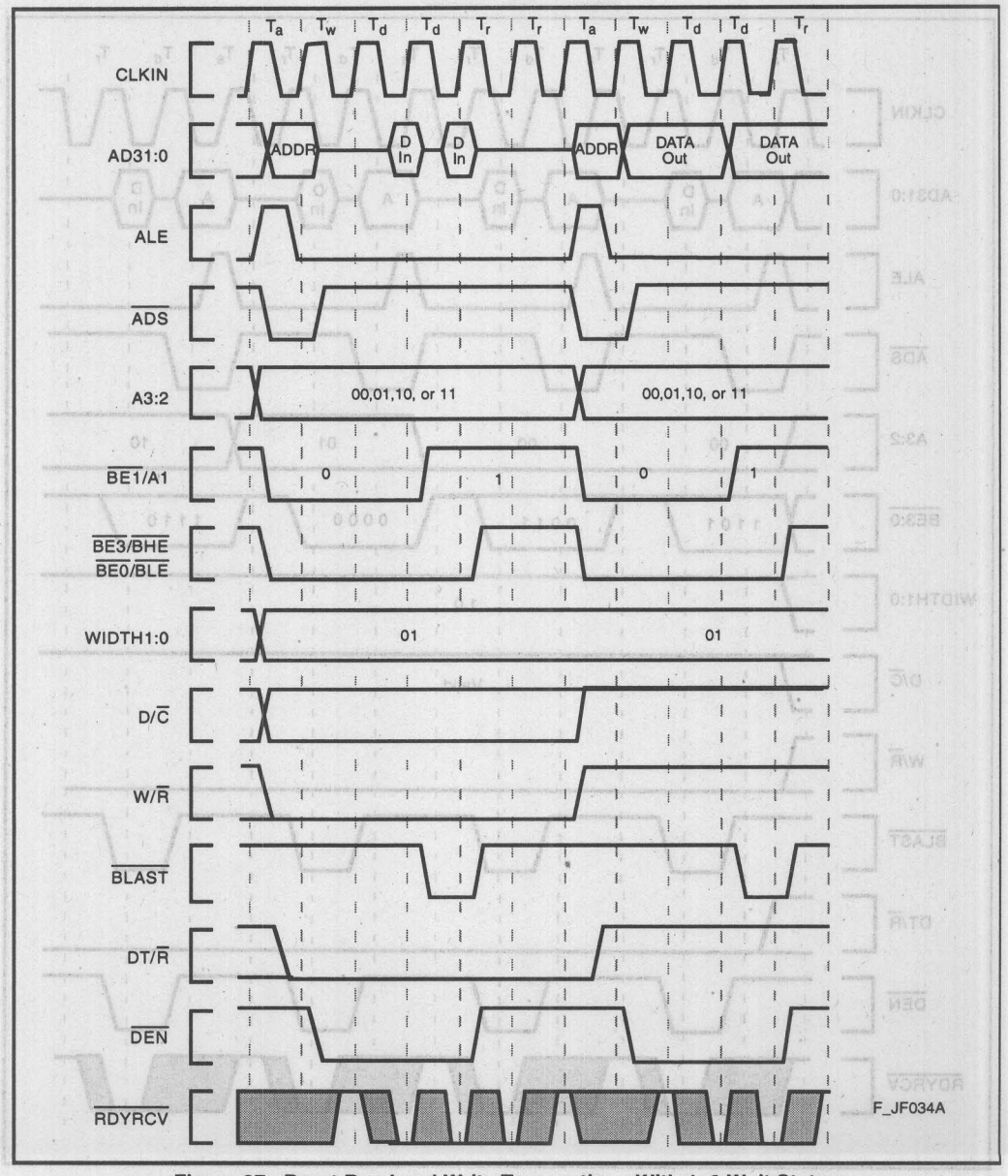


Figure 27. Burst Read and Write Transactions With 1, 0 Wait States and Extra Tr State on Read, 16-Bit Bus

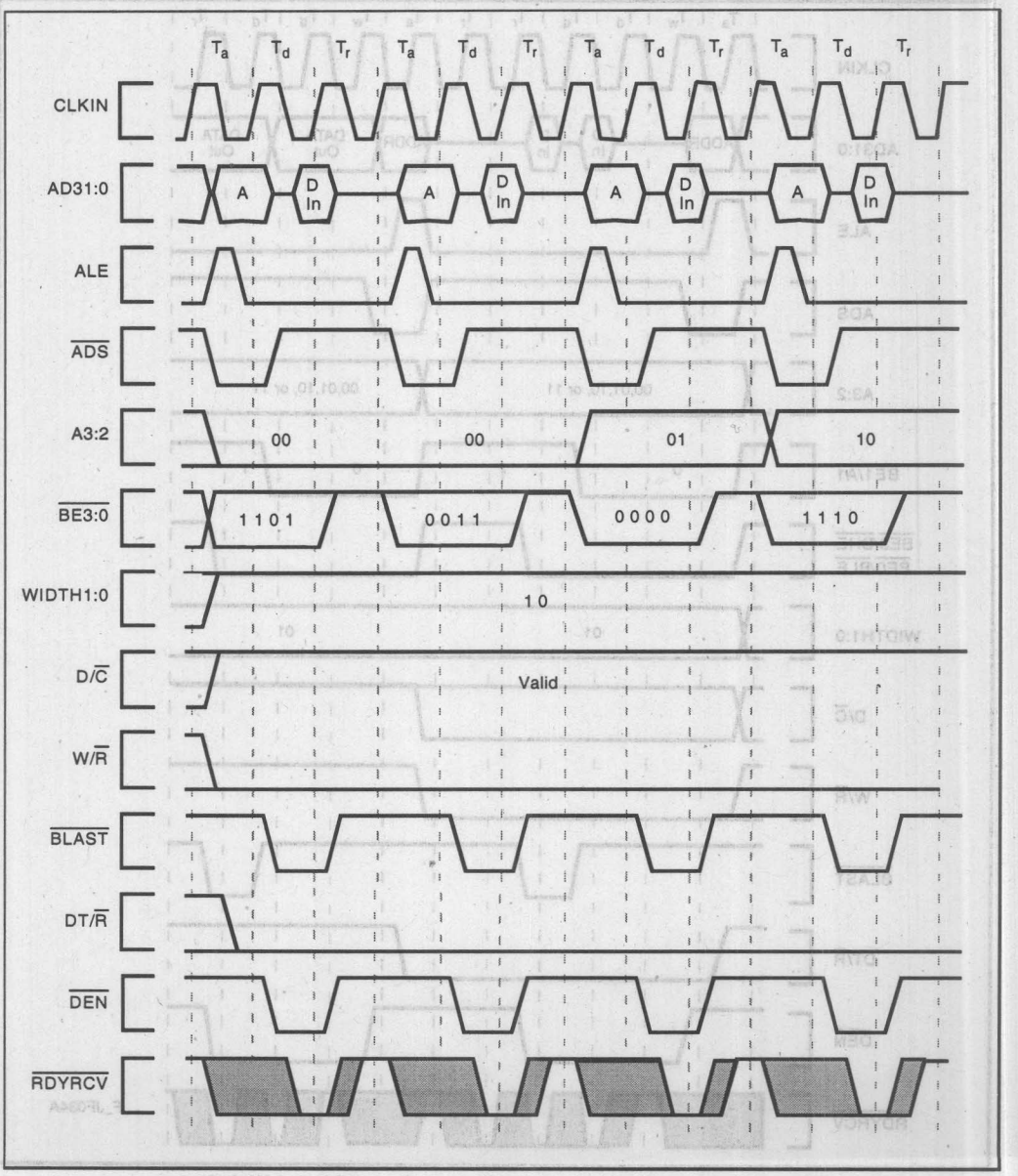
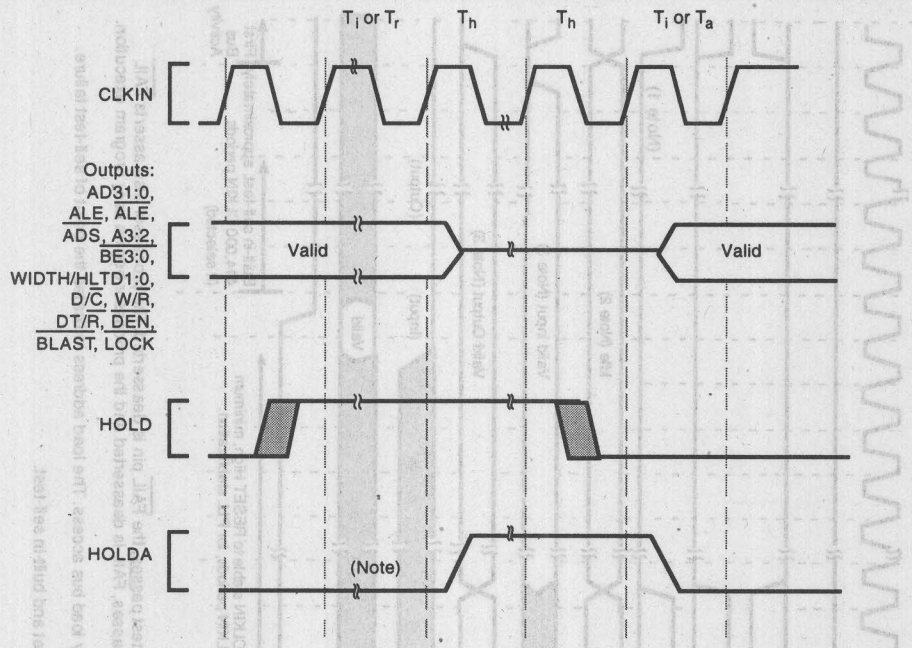


Figure 28. Bus Transactions Generated by Double Word Read Bus Request, Misaligned One Byte From Quad Word Boundary, 32-Bit Bus, Little Endian



NOTE: HOLD is sampled on the rising edge of CLKIN. The processor asserts HOLDA to grant the bus on the same edge in which it recognizes HOLD if the last state was T_i or the last T_r of a bus transaction. Similarly, the processor deasserts HOLDA on the same edge in which it recognizes the deassertion of HOLD.

Figure 29. HOLD/HOLDA Waveform For Bus Arbitration

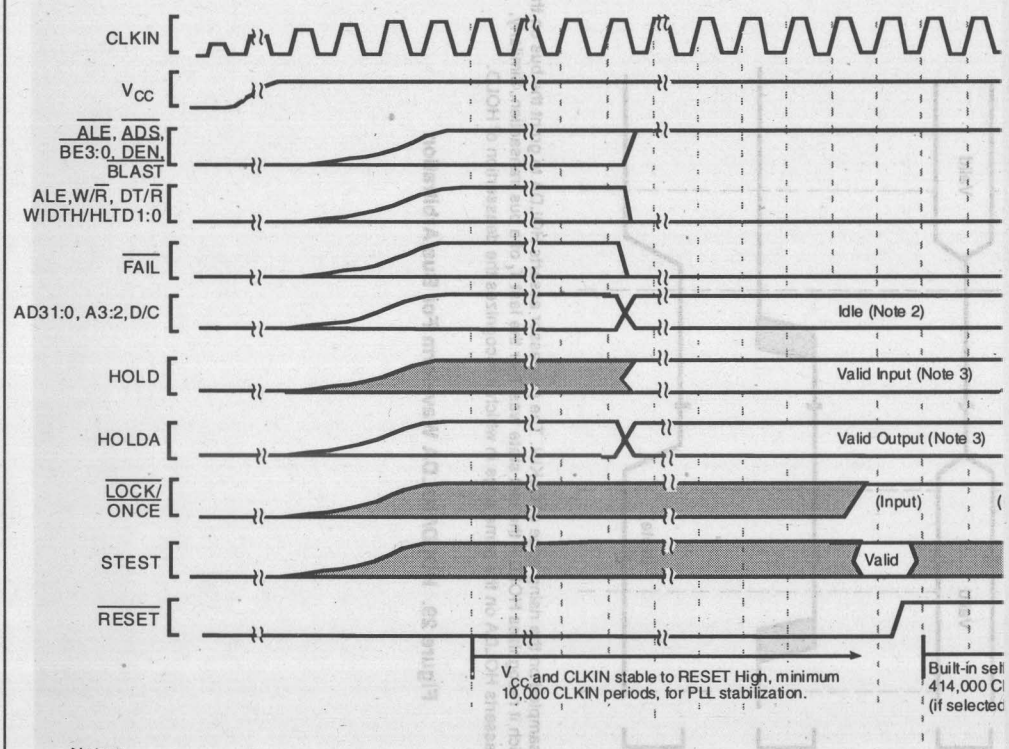


Figure 30. Cold Reset Waveform

Notes:

1. The processor asserts FAIL during built-in self-test. If self- test passes, the FAIL pin is deasserted. The processor during the bus confidence test. If the bus confidence test passes, FAIL is deasserted and the processor begins u

2. If the processor fails built-in self-test, it initiates one dummy load bus access. The load address indicates the poir

3. Since the bus is idle, hold requests are honored during reset and built-in self-test.

PRELIMINARY

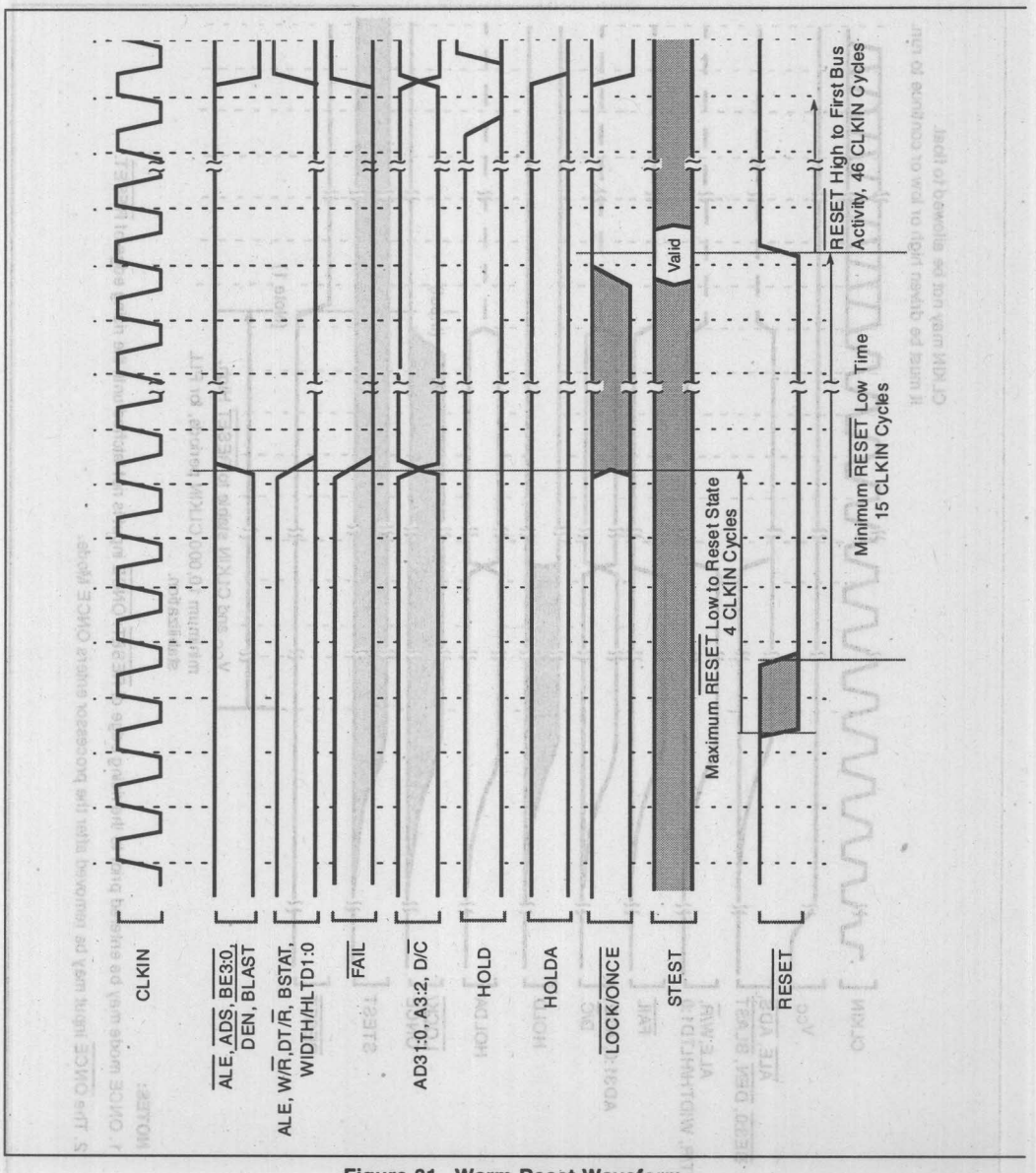
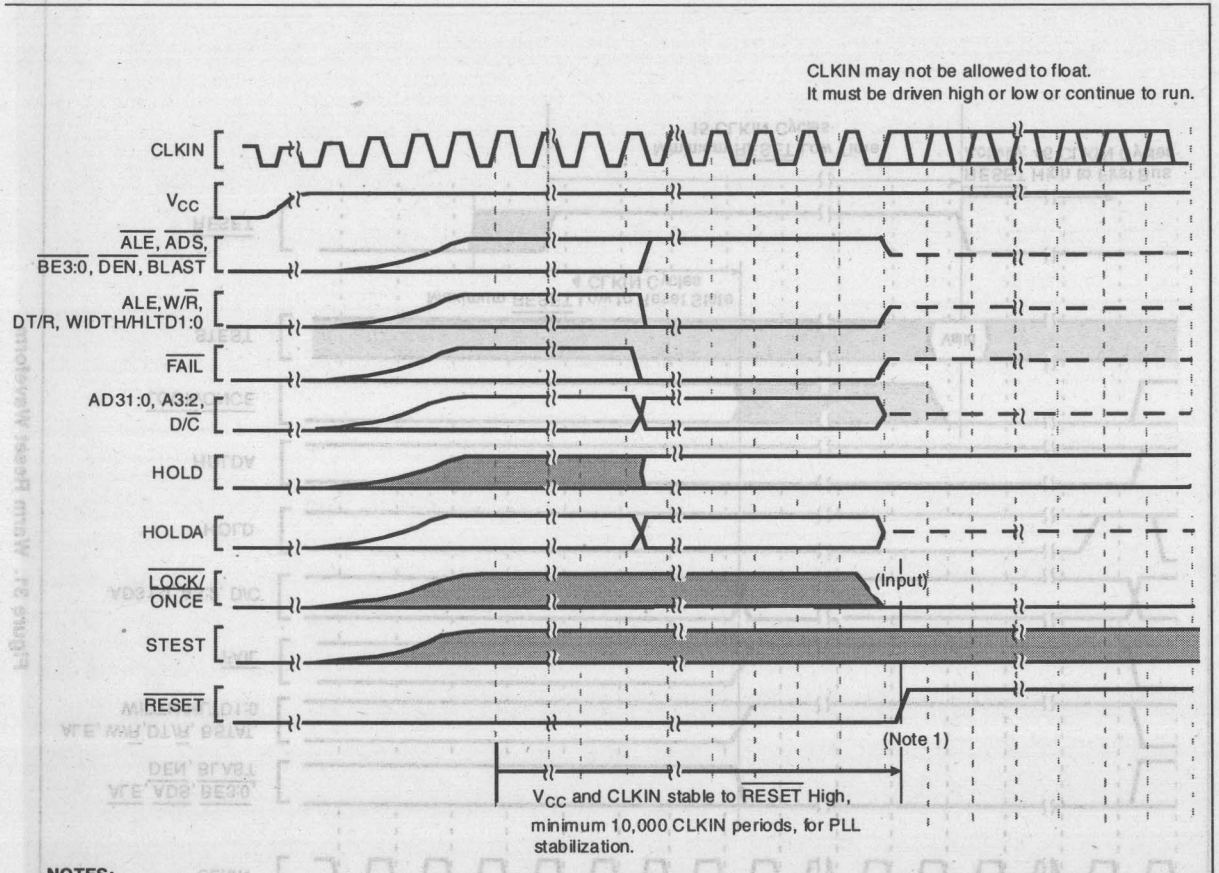


Figure 31. Warm Reset Waveform



NOTES:

1. ONCE mode may be entered prior to the rising edge of RESET: ONCE input is not latched until the rising edge of RESET.
2. The ONCE input may be removed after the processor enters ONCE Mode.

Figure 32. Entering the ONCE State

Double Word	8
Triple Word	16
Quad Word	16

Table 19. Summary of Byte Load and Store Accesses

Address Offset from Natural Boundary (in Bytes)	Accesses on 8-Bit Bus (WIDTH1:0=00)	Accesses on 16 Bit Bus (WIDTH1:0=01)	Accesses on 32 Bit Bus (WIDTH1:0=10)
+0 (aligned)	• byte access	• byte access	• byte access

Table 20. Summary of Short Word Load and Store Accesses

Address Offset from Natural Boundary (in Bytes)	Accesses on 8-Bit Bus (WIDTH1:0=00)	Accesses on 16 Bit Bus (WIDTH1:0=01)	Accesses on 32 Bit Bus (WIDTH1:0=10)
+0 (aligned)	• burst of 2 bytes	• short-word access	• short-word access
+1	• 2 byte accesses	• 2 byte accesses	• 2 byte accesses

		burst of 4 short words burst of 2 short words • case *n*=4: 2 bursts of 4 short words	
+1 (*n*=1, 2, 3, 4) +5 (*n* = 2, 3, 4) +9 (*n* = 3, 4) +13 (*n* = 3, 4)	• byte access • burst of 2 bytes • *n*-1 burst(s) of 4 bytes • byte access	• byte access • short-word access • *n*-1 burst(s) of 2 short words • byte access	• byte access • short-word access • *n*-1 word access(es) • byte access
+2 (*n*=1, 2, 3, 4) +6 (*n* = 2, 3, 4) +10 (*n* = 3, 4) +14 (*n* = 3, 4)	• burst of 2 bytes • *n*-1 burst(s) of 4 bytes • burst of 2 bytes	• short-word access • *n*-1 burst(s) of 2 short words • short-word access	• short-word access • *n*-1 word access(es) • short-word access
+3 (*n*=1, 2, 3, 4) +7 (*n* = 2, 3, 4) +11 (*n* = 3, 4) +15 (*n* = 3, 4)	• byte access • *n*-1 burst(s) of 4 bytes • burst of 2 bytes • byte access	• byte access • *n*-1 burst(s) of 2 short words • short-word access • byte access	• byte access • *n*-1 word access(es) • short-word access • byte access
+4 (*n* = 2, 3, 4) +8 (*n* = 3, 4) +12 (*n* = 3, 4)	• *n* burst(s) of 4 bytes	• *n* burst(s) of 2 short words	• *n* word access(es)

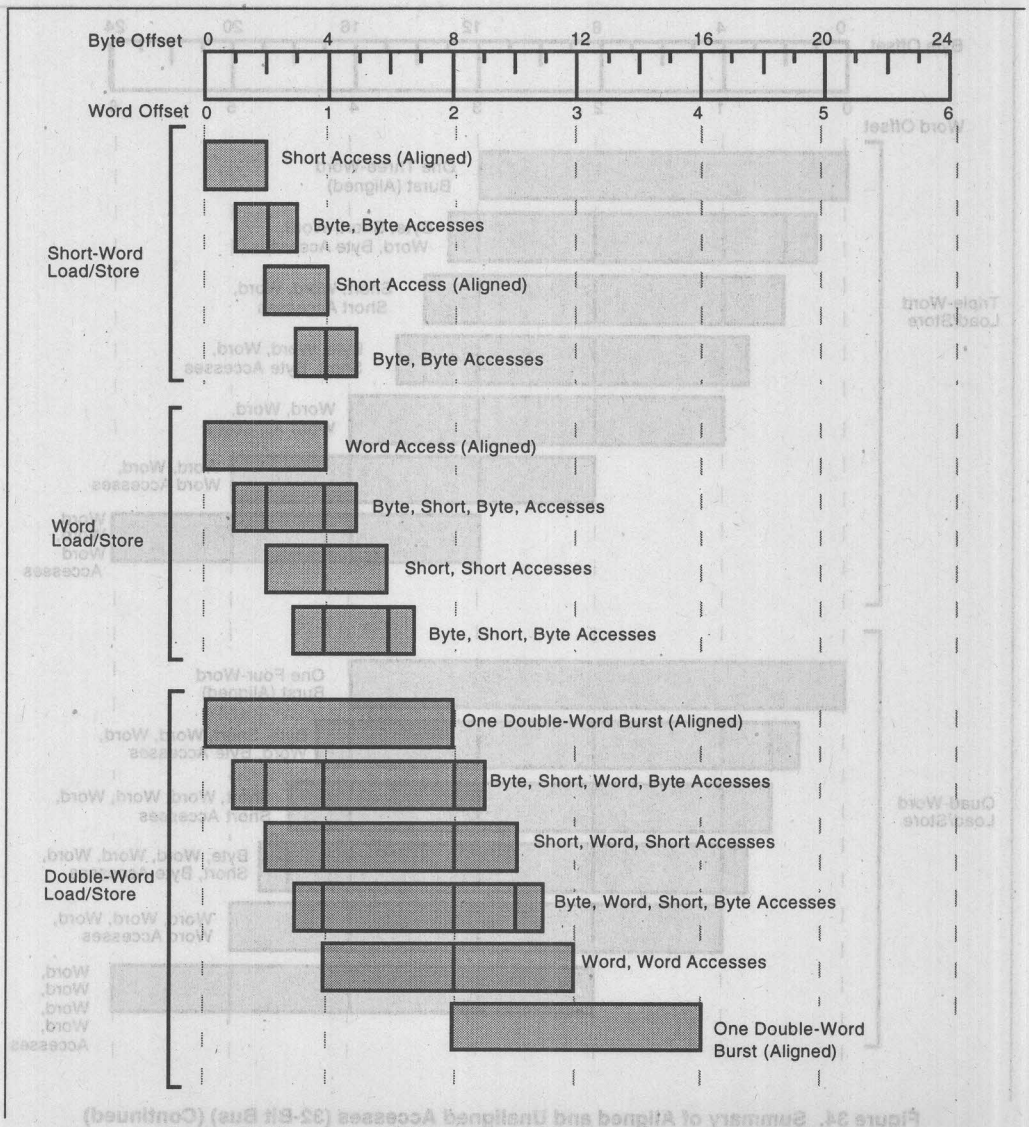


Figure 33. Summary of Aligned and Unaligned Accesses (32-Bit Bus)

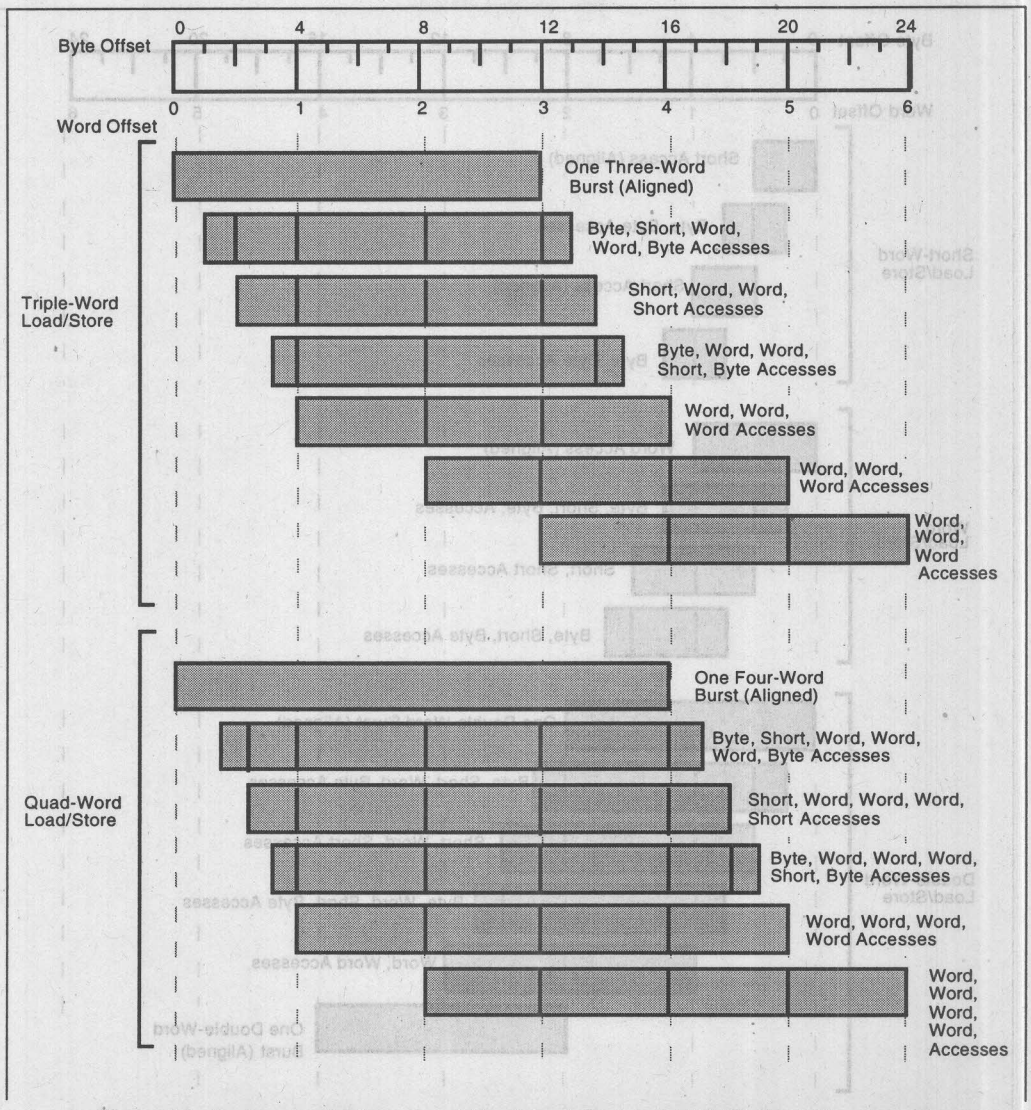


Figure 34. Summary of Aligned and Unaligned Accesses (32-Bit Bus) (Continued)

6.0 DEVICE IDENTIFICATION

80L960JA/JF processors may be identified electrically according to device type and stepping (see Table 22). The 32-bit identifier is accessible in three ways:

- Upon reset, the identifier is placed into the g0 register.
- The identifier may be accessed from supervisor mode at any time by reading the DEVICEID register at address FF008710H.
- The IEEE Standard 1149.1 Test Access Port may select the DEVICE ID register through the IDCODE instruction.

The device and stepping letter is also printed on the top side of the product package.

Table 22. 80L960JA/JF Die and Stepping Reference

Device and Stepping	Version Number	Part Number	Manufacturer	X	Complete ID (Hex)
80L960JA A, A2	0000	0000 1000 0010 0001	0000 0001 001	1	00821013
80L960JF A, A2	0000	0000 1000 0010 0000	0000 0001 001	1	00820013

NOTE: This data sheet applies to the 80L960JA/JF A and 80L960JA/JF A2 steppings.

7.0 REVISION HISTORY

This data sheet supersedes revision 272744-001. Table 23 indicates significant changes since the previous revision.

Table 23. Data Sheet Version -001 to -002 Revision History (Sheet 1 of 2)

Table 14, 80L960JA/JF ICC Characteristics (pg. 24)	I_{CC} Active (Power Supply): 80L960JA/J F-25 improved from 330 mA to 284 mA, 80L960JA/J F-16 improved from 220 mA to 184 mA. I_{CC} Active (Thermal) 80L960JA/J F-25 changed from 225 mA to 255 mA. I_{CC} Test (Power Modes): added Reset I_{CC} values, 80L960JA/J F-16 improved from 21 mA to 16 mA, ONCE mode improved from 12 mA to 10mA.
Section 4.5, AC Specifications (pg. 25)	Grouped AC Specifications tables by frequency. Added 16 MHz AC specifications.
Table 15, 80L960JA/JF AC Characteristics (25 MHz) (pg. 25) Section SYNCHRONOUS OUTPUT TIMINGS	T_{ov1} (min) improved from 4.5 ns to 4.0 ns. T_{ov1} (max) improved from 19 ns to 18 ns. T_{ov2} (max) improved from $0.5T_C$ + 16 ns to $0.5T_C$ + 18 ns. T_{oF} (min) improved from 4.5 ns to 4.0 ns. T_{oF} (max) improved from 13 ns to 16 ns.
Table 15, 80L960JA/JF AC Characteristics (25 MHz) (pg. 25) Section SYNCHRONOUS INPUT TIMINGS	T_{IS1} (min) improved from 10 ns to 9 ns. T_{IH1} (min) improved from 3 ns to 1 ns. T_{IS2} (min) improved from 11 ns to 10 ns. T_{IH2} (min) improved from 3 ns to 1 ns. T_{IS3} (min) improved from 9 ns to 8 ns. T_{IH3} (min) improved from 3 ns to 1 ns. T_{IS4} (min) improved from 9 ns to 8 ns. T_{IH4} (min) improved from 3 ns to 1 ns.
Table 15, 80L960JA/JF AC Characteristics (25 MHz) (pg. 25) Section RELATIVE OUTPUT TIMINGS	T_{LXL}, T_{LXA}, and T_{DXD} (min) improved from $.5T_C$ - 6.5 ns to $.5T_C$ - 7.5 ns.

Table 23. Data Sheet Version -001 to -002 Revision History (Sheet 2 of 2)

Table 15, 80L960JA/JF AC Characteristics (25 MHz) (pg. 25) Section BOUNDARY SCAN TEST SIGNAL TIMINGS	T_{BSIS1} (min) improved from 8 ns to 4 ns. T_{BSIH1} (min) improved from 10 ns to 6 ns. T_{BSOV2} (max) improved from 30 ns to 35 ns. T_{BSIS2} (min) improved from 8 ns to 4 ns. T_{BSIH2} (min) improved from 10 ns to 6 ns.

- — Sixteen 32-Bit Local Registers (8 sets)
- — Nine Addressing Modes
- — User/Supervisor Protection Model

- Two-Way Set Associative Instruction Cache
  - — 80960JD - 4 Kbyte
  - — Programmable Cache Locking Mechanism
- Direct Mapped Data Cache
  - — 80960JD - 2 Kbyte
  - — Write Through Operation
- On-Chip Stack Frame Cache
  - — Seven Register Sets Can Be Saved
  - — Automatic Allocation on Call/Return
  - — 0-7 Frames Reserved for High-Priority Interrupts
- On-Chip Data RAM
  - — 1 Kbyte Critical Variable Storage
  - — Single-Cycle Access
  - — Conditional Add, Subtract and Select
  - — Processor Management
- High-Speed Interrupt Controller
  - — 31 Programmable Priorities
  - — Eight Maskable Pins plus NMI
  - — Up to 240 Vectors in Expanded Mode
- Two On-Chip Timers
  - — Independent 32-Bit Counting
  - — Clock Prescaling by 1, 2, 4 or 8
  - — Internal Interrupt Sources
- Halt Mode for Low Power
- IEEE 1149.1 (JTAG) Boundary Scan Compatibility
- Packages
  - — 132-Lead Pin Grid Array (PGA)
  - — 132-Lead Plastic Quad Flat Pack (PQFP)

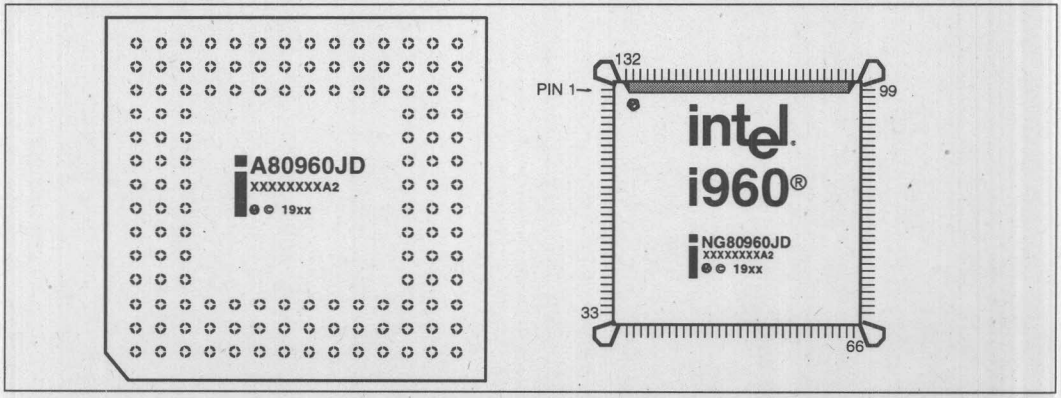


Figure 1. 80960JD Microprocessor

intel

PRELIMINARY

80960JD

EMBEDDED 32-BIT MICROPROCESSOR

- Pin/Code Compatible with all 80960Jx Processors
- High-Performance Embedded Architecture
  - One Instruction/Clock Execution
  - Core Clock Rate is 2x the Bus Clock
  - Load/Store Programming Model
  - Sixteen 32-Bit Global Registers
  - Sixteen 32-Bit Local Registers (8 sets)
  - Nine Addressing Modes
  - User/Supervisor Protection Model
- Two-Way Set Associative Instruction Cache
  - 80960JD - 4 Kbyte
  - Programmable Cache Locking Mechanism
- Direct Mapped Data Cache
  - 80960JD - 2 Kbyte
  - Write Through Operation
- On-Chip Stack Frame Cache
  - Seven Register Sets Can Be Saved
  - Automatic Allocation on Call/Return
  - 0-7 Frames Reserved for High-Priority Interrupts
- On-Chip Data RAM
  - 1 Kbyte Critical Variable Storage
  - Single-Cycle Access
- High Bandwidth Burst Bus
  - 32-Bit Multiplexed Address/Data
  - Programmable Memory Configuration
  - Selectable 8-, 16-, 32-Bit Bus Widths
  - Supports Unaligned Accesses
  - Big or Little Endian Byte Ordering
- New Instructions
  - Conditional Add, Subtract and Select
  - Processor Management
- High-Speed Interrupt Controller
  - 31 Programmable Priorities
  - Eight Maskable Pins plus NMI
  - Up to 240 Vectors in Expanded Mode
- Two On-Chip Timers
  - Independent 32-Bit Counting
  - Clock Prescaling by 1, 2, 4 or 8
  - Internal Interrupt Source
- Halt Mode for Low Power
- IEEE 1149.1 (JTAG) Boundary Scan Compatibility
- Packages
  - 132-Lead Pin Grid Array (PGA)
  - 132-Lead Plastic Quad Flat Pack (PQFP)

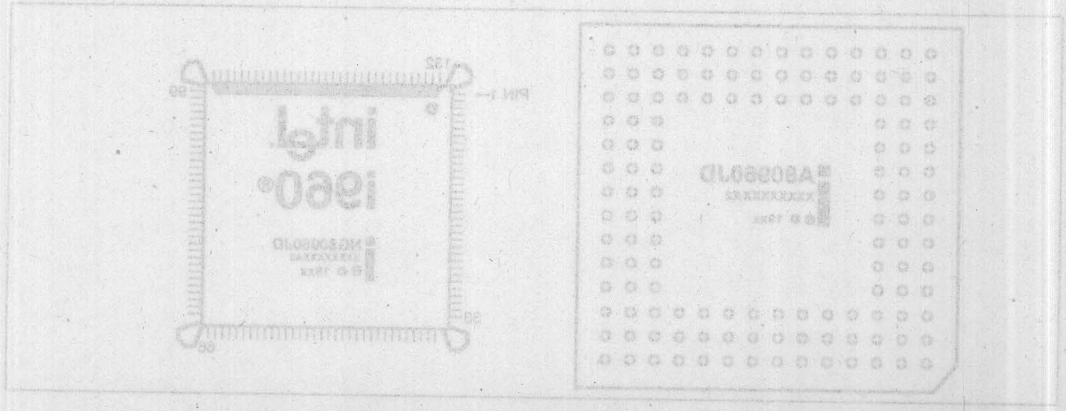


Figure 1. 80960JD Microprocessor

1.0 PURPOSE

This document contains advance information for the 80960JD microprocessor, including electrical characteristics and package pinout information. Detailed functional descriptions — other than parametric performance — are published in the *i960® Jx Microprocessor User's Guide* (272483).

Throughout this data sheet, references to "80960Jx" indicate features which apply to all of the following:

- 80960JA — 5V, 2 Kbyte instruction cache, 1 Kbyte data cache
- 80960JF — 5V, 4 Kbyte instruction cache, 2 Kbyte data cache
- 80960JD — 5V, 4 Kbyte instruction cache, 2 Kbyte data cache and clock doubling
- 80L960JA — 3.3 V version of the 80960JA
- 80L960JF — 3.3 V version of the 80960JF

2.0 80960JD OVERVIEW

The 80960JD offers high performance to cost-sensitive 32-bit embedded applications. The 80960JD is object code compatible with the 80960 Core Architecture and is capable of sustained execution at the rate of one instruction per clock. This processor's features include generous instruction cache, data cache and data RAM. It also boasts a fast interrupt mechanism, dual programmable timer units and new instructions.

The 80960JD's clock doubler operates the processor core at twice the bus clock rate to improve execution performance without increasing the complexity of board designs.

Memory subsystems for cost-sensitive embedded applications often impose substantial wait state penalties. The 80960JD integrates considerable storage resources on-chip to decouple CPU execution from the external bus.

The 80960JD rapidly allocates and deallocates local register sets during context switches. The processor needs to flush a register set to the stack only when it saves more than seven sets to its local register cache.

A 32-bit multiplexed burst bus provides a high-speed interface to system memory and I/O. A full complement of control signals simplifies the connection of the 80960JD to external components. The user programs physical and logical memory attributes through memory-mapped control registers (MMRs) — an extension not found on the i960 Kx, Sx or Cx processors. Physical and logical configuration registers enable the processor to operate with all combinations of bus width and data object alignment. The processor supports a homogeneous byte ordering model.

This processor integrates two important peripherals: a timer unit and an interrupt controller. These and other hardware resources are programmed through memory-mapped control registers, an extension to the familiar 80960 architecture.

The timer unit (TU) offers two independent 32-bit timers for use as real-time system clocks and general-purpose system timing. These operate in either single-shot or auto-reload mode and can generate interrupts.

The interrupt controller unit (ICU) provides a flexible means for requesting interrupts. The ICU provides full programmability of up to 240 interrupt sources into 31 priority levels. The ICU takes advantage of a cached priority table and optional routine caching to minimize interrupt latency. Clock doubling reduces interrupt latency by 40% compared to the 80960JA/JF. Local registers may be dedicated to high-priority interrupts to further reduce latency. Acting independently from the core, the ICU compares the priorities of posted interrupts with the current process priority, off-loading this task from the core. The ICU also supports the integrated timer interrupts.

The 80960JD features a Halt mode designed to support applications where low power consumption is critical. The **halt** instruction shuts down instruction execution, resulting in a power savings of up to 90 percent.

The 80960JD's testability features, including ONCE (On-Circuit Emulation) mode and Boundary Scan (JTAG), provide a powerful environment for design debug and fault diagnosis.

The *Solutions960®* program features a wide variety of development tools which support the i960 processor family. Many of these tools are developed by partner companies; some are developed by Intel, such as profile-driven optimizing compilers. For more information on these products, contact your local Intel representative.

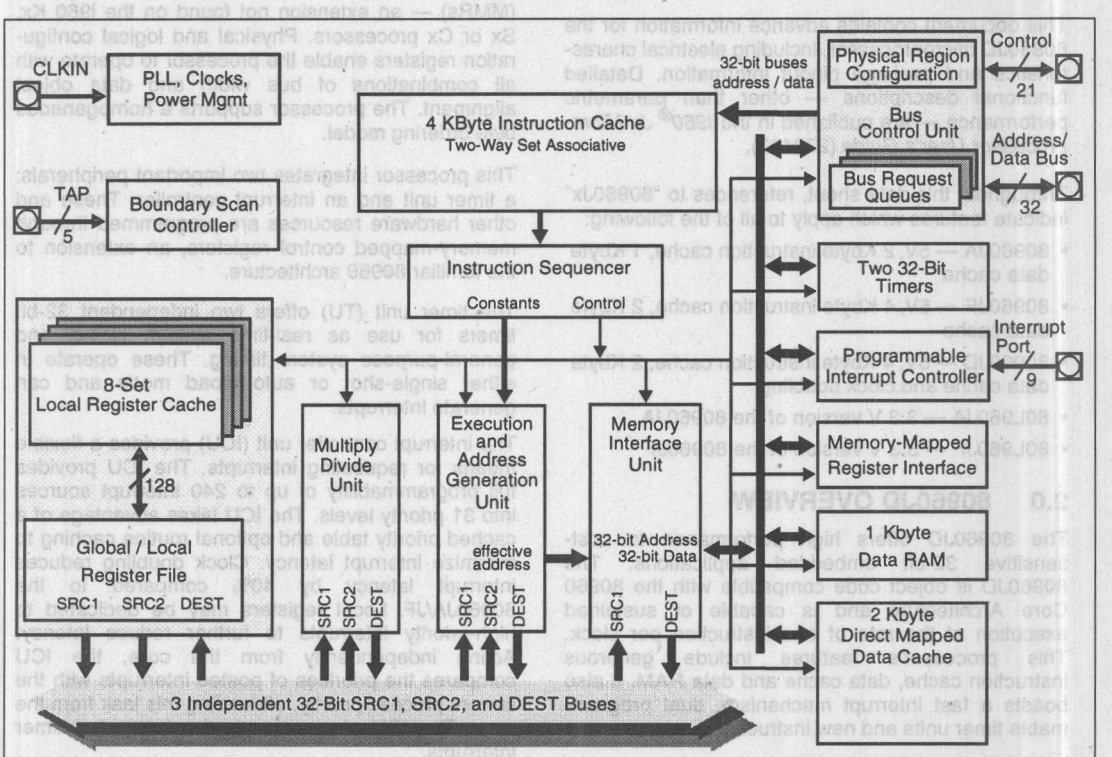


Figure 2. 80960JD Block Diagram

2.1 80960 Processor Core

The 80960Jx family is a scalar implementation of the 80960 Core Architecture. Intel designed this processor core as a very high performance device that is also cost-effective. Factors that contribute to the core's performance include:

- Core operates at twice the bus speed (80960JD only)
- Single-clock execution of most instructions
- Independent Multiply/Divide Unit
- Efficient instruction pipeline minimizes pipeline break latency
- Register and resource scoreboarding allow overlapped instruction execution
- 128-bit register bus speeds local register caching
- 4 Kbyte two-way set associative, integrated instruction cache
- 2 Kbyte direct-mapped, integrated data cache
- 1 Kbyte integrated data RAM delivers zero wait state program data

2.2 Burst Bus

A 32-bit high-performance bus controller interfaces the 80960JD to external memory and peripherals. The BCU fetches instructions and transfers data at the rate of up to four 32-bit words per six clock cycles. The external address/data bus is multiplexed.

Users may configure the 80960JD's bus controller to match an application's fundamental memory organization. Physical bus width is register-programmed for up to eight regions. Byte ordering and data caching are programmed through a group of logical memory templates and a defaults register.

The BCU's features include:

- Multiplexed external bus to minimize pin count
- 32-, 16- and 8-bit bus widths to simplify I/O interfaces
- External ready control for address-to-data, data-to-data and data-to-next-address wait state types
- Support for big or little endian byte ordering to facilitate the porting of existing program code
- Unaligned bus accesses performed transparently
- Three-deep load/store queue to decouple the bus from the core

Upon reset, the 80960JD conducts an internal self test. Then, before executing its first instruction, it performs an external bus confidence test by performing a checksum on the first words of the initialization boot record (IBR).

The user may examine the contents of the caches at any time by executing special cache control instructions.

2.3 Timer Unit

The timer unit (TU) contains two independent 32-bit timers which are capable of counting at several clock rates and generating interrupts. Each is programmed by use of the TU registers. These memory-mapped registers are addressable on 32-bit boundaries. The timers have a single-shot mode and auto-reload capabilities for continuous operation. Each timer has an independent interrupt request to the 80960JD's interrupt controller. The TU can generate a fault when unauthorized writes from user mode are detected. Clock prescaling is supported.

2.4 Priority Interrupt Controller

A programmable interrupt controller manages up to 240 external sources through an 8-bit external interrupt port. Alternatively, the interrupt inputs may be configured for individual edge- or level-triggered inputs. The interrupt unit (IU) also accepts interrupts from the two on-chip timer channels and a single Non-Maskable Interrupt ($\overline{NMI}$) pin. Interrupts are serviced according to their priority levels relative to the current process priority.

Low interrupt latency is critical to many embedded applications. As part of its highly flexible interrupt mechanism, the 80960JD exploits several techniques to minimize latency:

- Interrupt vectors and interrupt handler routines can be reserved on-chip
- Register frames for high-priority interrupt handlers can be cached on-chip
- The interrupt stack can be placed in cacheable memory space
- Interrupt microcode executes at twice the bus frequency

2.5 Instruction Set Summary

The 80960Jx adds several new instructions to the i960 core architecture. The new instructions are:

- Conditional Move
- Conditional Add
- Conditional Subtract
- Byte Swap
- Halt
- Cache Control
- Interrupt Control

Table 1 identifies the instructions that the 80960Jx supports. Refer to *i960® Jx Microprocessor User's Guide* (272483) for a detailed description of each instruction.

2.6 Faults and Debugging

The 80960Jx employs a comprehensive fault model. The processor responds to faults by making implicit calls to a fault handling routine. Specific information collected for each fault allows the fault handler to diagnose exceptions and recover appropriately.

The processor also has built-in debug capabilities. In software, the 80960Jx may be configured to detect as many as seven different trace event types. Alter-

Intel fabricates the 80960Jx using an advanced sub-micron manufacturing process. The processor's sub-micron topology provides the circuit density for optimal cache size and high operating speeds while dissipating modest power. The processor also uses dynamic power management to turn off clocks to unused circuits.

Users may program the 80960Jx to enter Halt mode for maximum power savings. In Halt mode, the processor core stops completely while the integrated peripherals continue to function, reducing overall power requirements up to 90 percent. Processor execution resumes from internally or externally generated interrupts.

2.8 Test Features

The 80960Jx incorporates numerous features which enhance the user's ability to test both the processor and the system to which it is attached. These features include ONCE (On-Circuit Emulation) mode and Boundary Scan (JTAG).

The 80960Jx provides testability features compatible with IEEE Standard Test Access Port and Boundary Scan Architecture (IEEE Std. 1149.1).

One of the boundary scan instructions, HIGHZ, forces the processor to float all its output pins (ONCE mode). ONCE mode can also be initiated at reset without using the boundary scan mechanism.

ONCE mode is useful for board-level testing. This feature allows a mounted 80960JD to electrically "remove" itself from a circuit board. This allows for system-level testing where a remote tester — such as an in-circuit emulator — can exercise the processor system.

The provided test logic does not interfere with component or circuit board behavior and ensures that components function correctly, connections

2.9 Memory-Mapped Control Registers

The 80960JD, though compliant with i960 series processor core, has the added advantage of memory-mapped, internal control registers not found on the i960 Kx, Sx or Cx processors. These give software the interface to easily read and modify internal control registers.

Each of these registers is accessed as a memory-mapped, 32-bit register. Access is accomplished through regular memory-format instructions. The processor ensures that these accesses do not generate external bus cycles.

2.10 Data Types and Memory Addressing Modes

As with all i960 family processors, the 80960Jx instruction set supports several data types and formats:

- Bit
- Bit fields
- Integer (8-, 16-, 32-, 64-bit)
- Ordinal (8-, 16-, 32-, 64-bit unsigned integers)
- Triple word (96 bits)
- Quad word (128 bits)

The 80960Jx provides a full set of addressing modes for C and assembly programming:

- Two Absolute modes
- Five Register Indirect modes
- Index with displacement
- IP with displacement

Table 1. 80960Jx Instruction Set

Data Movement	Arithmetic	Logical	Bit, Bit Field and Byte
Load	Add	And	Set Bit
Store	Subtract	Not And	Clear Bit
Move	Multiply	And Not	Not Bit
*Conditional Select	Divide	Or	Alter Bit
Load Address	Remainder	Exclusive Or	Scan For Bit
	Modulo	Not Or	Span Over Bit
	Shift	Or Not	Extract
	Extended Shift	Nor	Modify
	Extended Multiply	Exclusive Nor	Scan Byte for Equal
	Extended Divide	Not	*Byte Swap
	Add with Carry	Nand	
	Subtract with Carry		
	*Conditional Add		
	*Conditional Subtract		
	Rotate		

Comparison	Branch	Call/Return	Fault
Compare	Unconditional Branch	Call	Conditional Fault
Conditional Compare	Conditional Branch	Call Extended	Synchronize Faults
Compare and Increment	Compare and Branch	Call System	
Compare and Decrement		Return	
Test Condition Code		Branch and Link	
Check Bit			

Debug	Processor Management	Atomic
Modify Trace Controls	Flush Local Registers	Atomic Add
Mark	Modify Arithmetic Controls	Atomic Modify
Force Mark	Modify Process Controls	
	*Halt	
	System Control	
	*Cache Control	
	*Interrupt Control	

NOTE: Asterisk (*) denotes new 80960Jx instructions unavailable on 80960CA/CF, 80960KA/KB and 80960SA/SB implementations.

3.0 PACKAGE INFORMATION

The 80960JD is offered in several speed and package types. The 132-pin Pin Grid Array (PGA) device will be specified for operation at V_{CC} = 5.0 V ± 5% over a case temperature range of 0° to 85°C:

- A80960JD-50 (50 MHz core, 25 MHz bus)

The 132-pin Pin Grid Array (PGA) device will be specified for operation at V_{CC} = 5.0 V ± 5% over a case temperature range of 0° to 100°C:

- A80960JD-40 (40 MHz core, 20 MHz bus)
- A80960JD-33 (33.33 MHz core, 16.67 MHz bus)

The 132-pin Plastic Quad Flatpack (PQFP) devices will be specified for operation at V_{CC} = 5.0 V ± 5% over a case temperature range of 0° to 100°C:

- NG80960JD-40 (40 MHz core, 20 MHz bus)
- NG80960JD-33 (33.33 MHz core, 16.67 MHz bus)

For complete package specifications and information, refer to Intel's Packaging Handbook (240800).

3.1 Pin Descriptions

This section describes the pins for the 80960JD in the 132-pin ceramic Pin Grid Array (PGA) package and 132-lead Plastic Quad Flatpack Package (PQFP).

Section 3.1.1, Functional Pin Definitions describes pin function; **Section 3.1.2, 80960Jx 132-Lead PGA Pinout** and **Section 3.1.3, 80960Jx PQFP Pinout** define the signal and pin locations for the supported package types.

3.1.1 Functional Pin Definitions

Table 2 presents the legend for interpreting the pin descriptions which follow. Pins associated with the bus interface are described in Table 3. Pins associated with basic control and test functions are described in Table 4. Pins associated with the Interrupt Unit are described in Table 5.

Table 2. Pin Description Nomenclature

Symbol	Description
I	Input pin only.
O	Output pin only.
I/O	Pin can be either an input or output.
–	Pin must be connected as described.
S	Synchronous. Inputs must meet setup and hold times relative to CLKIN for proper operation. S(E) Edge sensitive input S(L) Level sensitive input
A (...)	Asynchronous. Inputs may be asynchronous relative to CLKIN. A(E) Edge sensitive input A(L) Level sensitive input
R (...)	While the processor's $\overline{\text{RESET}}$ pin is asserted, the pin: R(1) is driven to V_{CC} R(0) is driven to V_{SS} R(Q) is a valid output R(X) is driven to unknown state R(H) is pulled up to V_{CC}
H (...)	While the processor is in the hold state, the pin: H(1) is driven to V_{CC} H(0) is driven to V_{SS} H(Q) Maintains previous state or continues to be a valid output H(Z) Floats
P (...)	While the processor is halted, the pin: P(1) is driven to V_{CC} P(0) is driven to V_{SS} P(Q) Maintains previous state or continues to be a valid output

		values. SIZE, which comprises bits 0-1 of the AD lines during a T_a cycle, specifies the number of data transfers during the bus transaction. AD1 AD0 Bus Transfers 0 0 1 Transfer 0 1 2 Transfers 1 0 3 Transfers 1 1 4 Transfers When the processor enters Halt mode, if the previous bus operation was a: • write — AD31:2 are driven with the last data value on the AD bus. • read — AD31:4 are driven with the last address value on the AD bus; AD3:2 are driven with the value of A3:2 from the last data cycle. Typically, AD1:0 reflect the SIZE information of the last bus transaction (either instruction fetch or load/store) that was executed before entering Halt mode.
ALE	O R(0) H(Z) P(0)	**ADDRESS LATCH ENABLE** indicates the transfer of a physical address. ALE is asserted during a T_a cycle and deasserted before the beginning of the T_d state. It is active HIGH and floats to a high impedance state during a hold cycle (T_h).
$\overline{\text{ALE}}$	O R(1) H(Z) P(1)	**ADDRESS LATCH ENABLE** indicates the transfer of a physical address. $\overline{\text{ALE}}$ is the inverted version of ALE. This signal gives the 80960JD a high degree of compatibility with existing 80960Kx systems.
$\overline{\text{ADS}}$	O R(1) H(Z) P(1)	**ADDRESS STROBE** indicates a valid address and the start of a new bus access. The processor asserts $\overline{\text{ADS}}$ for the entire T_a cycle. External bus control logic typically samples $\overline{\text{ADS}}$ at the end of the cycle.
A3:2	O R(X) H(Z) P(Q)	**ADDRESS3:2** comprise a partial demultiplexed address bus. *32-bit memory accesses:* the processor asserts address bits A3:2 during T_a. The partial word address increments with each assertion of $\overline{\text{RDYRCV}}$ during a burst. *16-bit memory accesses:* the processor asserts address bits A3:1 during T_a with A1 driven on the $\overline{\text{BE1}}$ pin. The partial short word address increments with each assertion of $\overline{\text{RDYRCV}}$ during a burst. *8-bit memory accesses:* the processor asserts address bits A3:0 during T_a, with A1:0 driven on $\overline{\text{BE1:0}}$. The partial byte address increments with each assertion of $\overline{\text{RDYRCV}}$ during a burst.

		$\overline{BE3}$ enables data on AD31:24 $\overline{BE2}$ enables data on AD23:16 $\overline{BE1}$ enables data on AD15:8 $\overline{BE0}$ enables data on AD7:0 *16-bit bus:* $\overline{BE3}$ becomes Byte High Enable (enables data on AD15:8) $\overline{BE2}$ is not used (state is high) $\overline{BE1}$ becomes Address Bit 1 (A1) $\overline{BE0}$ becomes Byte Low Enable (enables data on AD7:0) *8-bit bus:* $\overline{BE3}$ is not used (state is high) $\overline{BE2}$ is not used (state is high) $\overline{BE1}$ becomes Address Bit 1 (A1) $\overline{BE0}$ becomes Address Bit 0 (A0) The processor asserts byte enables, byte high enable and byte low enable during T_a. Since unaligned bus requests are split into separate bus transactions, these signals do not toggle during a burst. They remain active through the last T_d cycle. For accesses to 8- and 16-bit memory, the processor asserts the address bits in conjunction with A3:2 described above.
WIDTH/ HLTD1:0	O R(0) H(Z) P(1)	**WIDTH/HALTED** signals denote the physical memory attributes for a bus transaction: WIDTH/HLTD1 WIDTH/HLTD0 0 0 8 Bits Wide 0 1 16 Bits Wide 1 0 32 Bits Wide 1 1 Processor Halted The processor floats the WIDTH/HLTD pins whenever it relinquishes the bus in response to a HOLD request, regardless of prior operating state.
D/$\overline{C}$	O R(X) H(Z) P(Q)	**DATA/CODE** indicates that a bus access is a data access (1) or an instruction access (0). D/$\overline{C}$ has the same timing as W/$\overline{R}$. 0 = instruction access 1 = data access
W/$\overline{R}$	O R(0) H(Z) P(Q)	**WRITE/READ** specifies, during a T_a cycle, whether the operation is a write (1) or read (0). It is latched on-chip and remains valid during T_d cycles. 0 = read 1 = write

Table 3. Pin Description — External Bus Signals (Sheet 3 of 4)

NAME	TYPE	DESCRIPTION
DT/$\overline{R}$	O R(0) H(Z) P(Q)	**DATA TRANSMIT / RECEIVE** indicates the direction of data transfer to and from the address/data bus. It is low during T_a and T_w/T_d cycles for a read; it is high during T_a and T_w/T_d cycles for a write. DT/$\overline{R}$ never changes state when $\overline{DEN}$ is asserted. 0 = receive 1 = transmit
$\overline{DEN}$	O R(1) H(Z) P(1)	**DATA ENABLE** indicates data transfer cycles during a bus access. $\overline{DEN}$ is asserted at the start of the first data cycle in a bus access and deasserted at the end of the last data cycle. $\overline{DEN}$ is used with DT/$\overline{R}$ to provide control for data transceivers connected to the data bus. 0 = data cycle 1 = not data cycle
$\overline{BLAST}$	O R(1) H(Z) P(1)	**BURST LAST** indicates the last transfer in a bus access. $\overline{BLAST}$ is asserted in the last data transfer of burst and non-burst accesses. $\overline{BLAST}$ remains active as long as wait states are inserted via the $\overline{RDYRCV}$ pin. $\overline{BLAST}$ becomes inactive after the final data transfer in a bus cycle. 0 = last data transfer 1 = not last data transfer
$\overline{RDYRCV}$	I S(L)	**READY/RECOVER** indicates that data on AD lines can be sampled or removed. If $\overline{RDYRCV}$ is not asserted during a T_d cycle, the T_d cycle is extended to the next cycle by inserting a wait state (T_w). 0 = sample data 1 = don't sample data The $\overline{RDYRCV}$ pin has another function during the recovery (T_r) state. The processor continues to insert additional recovery states until it samples the pin HIGH. This function gives slow external devices more time to float their buffers before the processor begins to drive address again. 0 = insert wait states 1 = recovery complete
$\overline{LOCK}$/ $\overline{ONCE}$	I/O S(L) R(H) H(Z) P(1)	**BUS LOCK** indicates that an atomic read-modify-write operation is in progress. The $\overline{LOCK}$ output is asserted in the first clock of an atomic operation and deasserted in the last data transfer of the sequence. The processor does not grant HOLDA while it is asserting $\overline{LOCK}$. This prevents external agents from accessing memory involved in semaphore operations. 0 = Atomic read-modify-write in progress 1 = Atomic read-modify-write not in progress **ONCE MODE:** The processor samples the $\overline{ONCE}$ input during reset. If it is asserted LOW at the end of reset, the processor enters ONCE mode. In ONCE mode, the processor stops all clocks and floats all output pins. The pin has a weak internal pullup which is active during reset to ensure normal operation when the pin is left unconnected. 0 = ONCE mode enabled 1 = ONCE mode not enabled

Table 3. Pin Description — External Bus Signals (Sheet 4 of 4)

NAME	TYPE	DESCRIPTION
HOLD	I S(L)	**HOLD**: A request from an external bus master to acquire the bus. When the processor receives HOLD and grants bus control to another master, it asserts HOLDA, floats the address/data and control lines and enters the T_h state. When HOLD is deasserted, the processor deasserts HOLDA and enters either the T_i or T_a state, resuming control of the address/data and control lines. 0 = no hold request 1 = hold request
HOLDA	O R(Q) H(1) P(Q)	**HOLD ACKNOWLEDGE** indicates to an external bus master that the processor has relinquished control of the bus. The processor can grant HOLD requests and enter the T_h state during reset and while halted as well as during regular operation. 0 = hold not acknowledged 1 = hold acknowledged
BSTAT	O R(0) H(Q) P(0)	**BUS STATUS** indicates that the processor may soon stall unless it has sufficient access to the bus; see *i960® Jx Microprocessor User's Guide* (272483). Arbitration logic can examine this signal to determine when an external bus master should acquire/relinquish the bus. 0 = no potential stall 1 = potential stall

Table 4. Pin Description — Processor Control Signals, Test Signals and Power (Sheet 1 of 2)

NAME	TYPE	DESCRIPTION
CLKIN	I	**CLOCK INPUT** provides the processor's fundamental time base; both the processor core and the external bus run at the CLKIN rate. All input and output timings are specified relative to a rising CLKIN edge.
$\overline{\text{RESET}}$	I A(L)	**RESET** initializes the processor and clears its internal logic. During reset, the processor places the address/data bus and control output pins in their idle (inactive) states. During reset, the input pins are ignored with the exception of $\overline{\text{LOCK}}$/$\overline{\text{ONCE}}$, STEST and HOLD. The $\overline{\text{RESET}}$ pin has an internal synchronizer. To ensure predictable processor initialization during power up, $\overline{\text{RESET}}$ must be asserted a minimum of 10,000 CLKIN cycles with V_{CC} and CLKIN stable. On a warm reset, $\overline{\text{RESET}}$ should be asserted for a minimum of 15 cycles.
STEST	I S(L)	**SELF TEST** enables or disables the processor's internal self-test feature at initialization. STEST is examined at the end of reset. When STEST is asserted, the processor performs its internal self-test and the external bus confidence test. When STEST is deasserted, the processor performs only the external bus confidence test. 0 = self test disabled 1 = self test enabled

Table 4. Pin Description — Processor Control Signals, Test Signals and Power (Sheet 2 of 2)

NAME	TYPE	DESCRIPTION
$\overline{\text{FAIL}}$	O R(0) H(Q) P(1)	**FAIL** indicates a failure of the processor's built-in self-test performed during initialization. $\overline{\text{FAIL}}$ is asserted immediately upon reset and toggles during self-test to indicate the status of individual tests: • When self-test passes, the processor deasserts $\overline{\text{FAIL}}$ and begins operation from user code. • When self-test fails, the processor asserts $\overline{\text{FAIL}}$ and then stops executing. 0 = self test failed 1 = self test passed
TCK	I	**TEST CLOCK** is a CPU input which provides the clocking function for IEEE 1149.1 Boundary Scan Testing (JTAG). State information and data are clocked into the processor on the rising edge; data is clocked out of the processor on the falling edge.
TDI	I S(L)	**TEST DATA INPUT** is the serial input pin for JTAG. TDI is sampled on the rising edge of TCK, during the SHIFT-IR and SHIFT-DR states of the Test Access Port.
TDO	O R(Q) HQ) P(Q)	**TEST DATA OUTPUT** is the serial output pin for JTAG. TDO is driven on the falling edge of TCK during the SHIFT-IR and SHIFT-DR states of the Test Access Port. At other times, TDO floats. TDO does not float during ONCE mode.
$\overline{\text{TRST}}$	I A(L)	**TEST RESET** asynchronously resets the Test Access Port (TAP) controller function of IEEE 1149.1 Boundary Scan testing (JTAG). When using the Boundary Scan feature, connect a pulldown resistor between this pin and V_{SS}. If TAP is not used, this pin must be connected to V_{SS}; however, no resistor is required. See Section 4.3, Connection Recommendations (pg. 24).
TMS	I S(L)	**TEST MODE SELECT** is sampled at the rising edge of TCK to select the operation of the test logic for IEEE 1149.1 Boundary Scan testing.
V_{CC}	–	**POWER** pins intended for external connection to a V_{CC} board plane.
V_{CCPLL}	–	**PLL POWER** is a separate V_{CC} supply pin for the phase lock loop clock generator. It is intended for external connection to the V_{CC} board plane. In noisy environments, add a simple bypass filter circuit to reduce noise-induced clock jitter and its effects on timing relationships.
V_{SS}	–	**GROUND** pins intended for external connection to a V_{SS} board plane.
NC	–	**NO CONNECT** pins. Do not make any system connections to these pins.

Table 5. Pin Description — Interrupt Unit Signals

NAME	TYPE	DESCRIPTION
$\overline{\text{XINT7:0}}$	I A(E/L)	**EXTERNAL INTERRUPT** pins are used to request interrupt service. The $\overline{\text{XINT7:0}}$ pins can be configured in three modes: **Dedicated Mode**: Each pin is assigned a dedicated interrupt level. Dedicated inputs can be programmed to be level (low) or edge (falling) sensitive. **Expanded Mode**: All eight pins act as a vectored interrupt source. The interrupt pins are level sensitive in this mode. **Mixed Mode**: The $\overline{\text{XINT7:5}}$ pins act as dedicated sources and the $\overline{\text{XINT4:0}}$ pins act as the five most significant bits of a vectored source. The least significant bits of the vectored source are set to 010_2 internally. Unused external interrupt pins should be connected to V_{CC}.
$\overline{\text{NMI}}$	I A(E)	**NON-MASKABLE INTERRUPT** causes a non-maskable interrupt event to occur. $\overline{\text{NMI}}$ is the highest priority interrupt source and is falling edge-triggered. If $\overline{\text{NMI}}$ is unused, it should be connected to V_{CC}.

3.1.2 80960Jx 132-Lead PGA Pinout

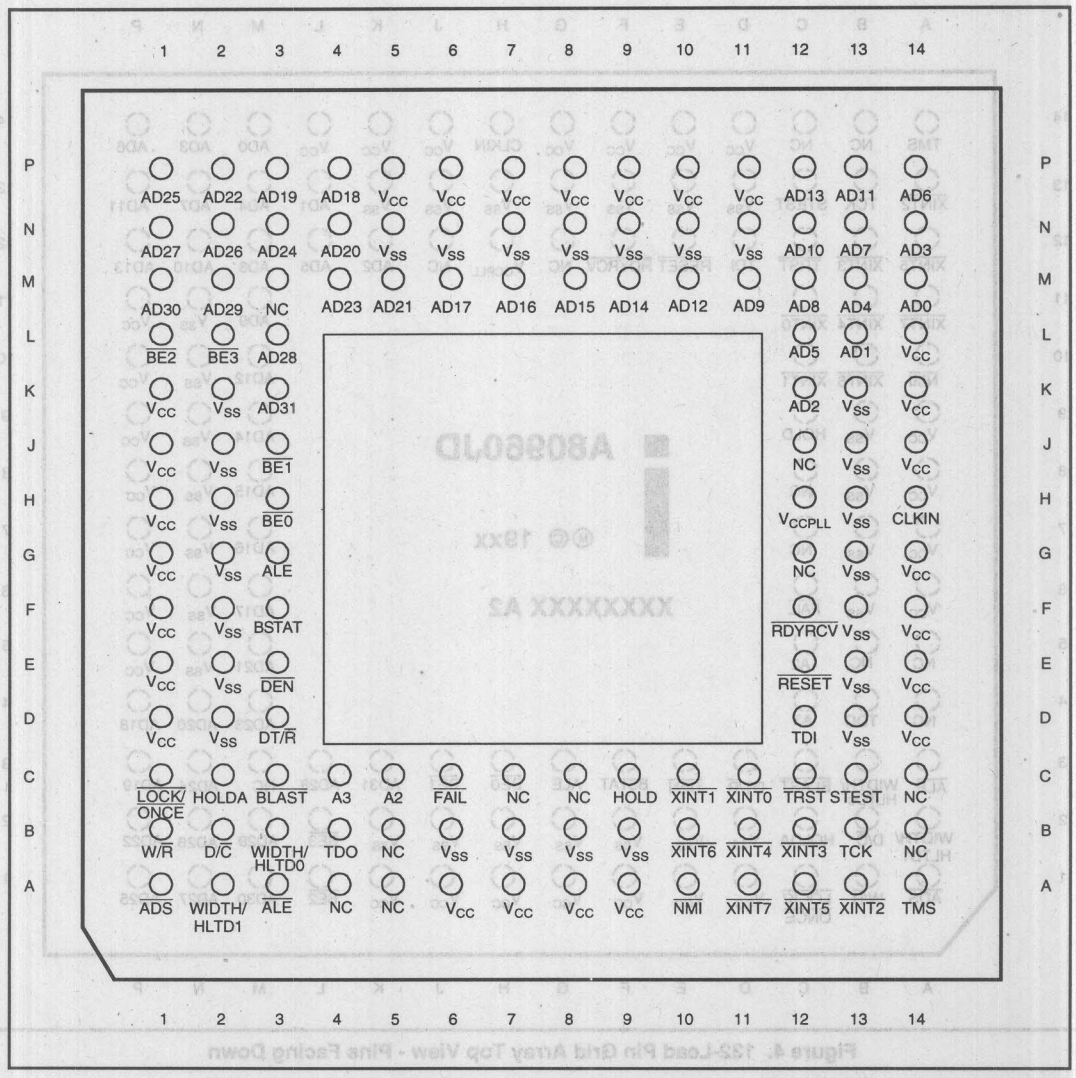


Figure 3. 132-Lead Pin Grid Array Bottom View - Pins Facing Up

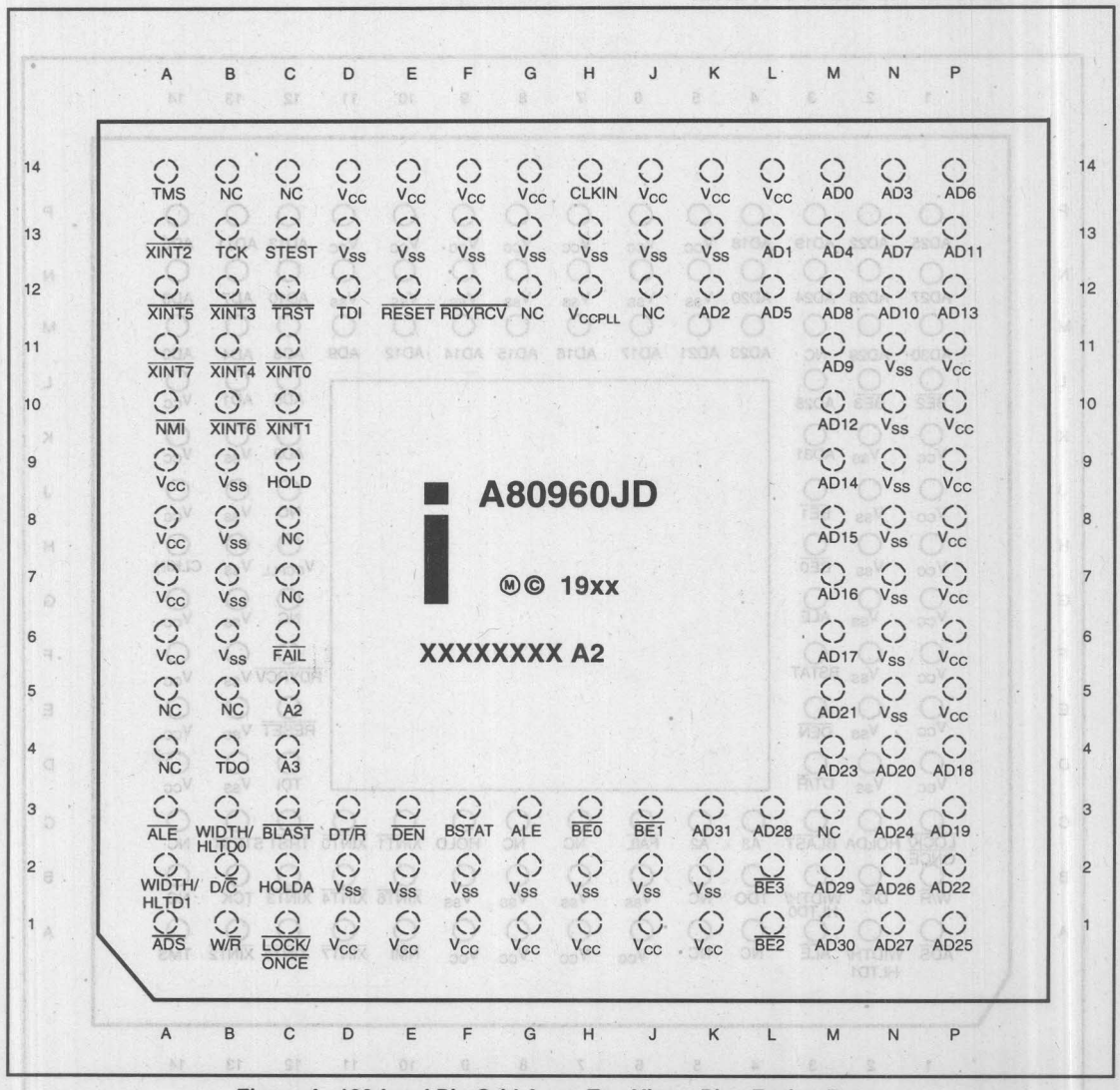


Figure 4. 132-Lead Pin Grid Array Top View - Pins Facing Down

Table 6. 132-Lead PGA Pinout — In Signal Order

Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin
A2	C5	AD31	K3	TDI	D12	V_{SS}	B9
A3	C4	$\overline{ADS}$	A1	TDO	B4	V_{SS}	D2
AD0	M14	ALE	G3	TMS	A14	V_{SS}	D13
AD1	L13	$\overline{ALE}$	A3	$\overline{TRST}$	C12	V_{SS}	E2
AD2	K12	$\overline{BE0}$	H3	V_{CC}	A6	V_{SS}	E13
AD3	N14	$\overline{BE1}$	J3	V_{CC}	A7	V_{SS}	F2
AD4	M13	$\overline{BE2}$	L1	V_{CC}	A8	V_{SS}	F13
AD5	L12	$\overline{BE3}$	L2	V_{CC}	A9	V_{SS}	G2
AD6	P14	$\overline{BLAST}$	C3	V_{CC}	D1	V_{SS}	G13
AD7	N13	BSTAT	F3	V_{CC}	D14	V_{SS}	H2
AD8	M12	CLKIN	H14	V_{CC}	E1	V_{SS}	H13
AD9	M11	D/$\overline{C}$	B2	V_{CC}	E14	V_{SS}	J2
AD10	N12	$\overline{DEN}$	E3	V_{CC}	F1	V_{SS}	J13
AD11	P13	DT/$\overline{R}$	D3	V_{CC}	F14	V_{SS}	K2
AD12	M10	$\overline{FAIL}$	C6	V_{CC}	G1	V_{SS}	K13
AD13	P12	HOLD	C9	V_{CC}	G14	V_{SS}	N5
AD14	M9	HOLDA	C2	V_{CC}	H1	V_{SS}	N6
AD15	M8	$\overline{LOCK/ONCE}$	C1	V_{CC}	J1	V_{SS}	N7
AD16	M7	NC	A4	V_{CC}	J14	V_{SS}	N8
AD17	M6	NC	A5	V_{CC}	K1	V_{SS}	N9
AD18	P4	NC	B5	V_{CC}	K14	V_{SS}	N10
AD19	P3	NC	B14	V_{CC}	L14	V_{SS}	N11
AD20	N4	NC	C7	V_{CC}	P5	W/$\overline{R}$	B1
AD21	M5	NC	C8	V_{CC}	P6	WIDTH/HLTD0	B3
AD22	P2	NC	C14	V_{CC}	P7	WIDTH/HLTD1	A2
AD23	M4	NC	G12	V_{CC}	P8	$\overline{XINT0}$	C11
AD24	N3	NC	J12	V_{CC}	P9	$\overline{XINT1}$	C10
AD25	P1	NC	M3	V_{CC}	P10	$\overline{XINT2}$	A13
AD26	N2	$\overline{NMI}$	A10	V_{CC}	P11	$\overline{XINT3}$	B12
AD27	N1	$\overline{RDYRCV}$	F12	V_{CCPLL}	H12	$\overline{XINT4}$	B11
AD28	L3	$\overline{RESET}$	E12	V_{SS}	B6	$\overline{XINT5}$	A12
AD29	M2	STEST	C13	V_{SS}	B7	$\overline{XINT6}$	B10
AD30	M1	TCK	B13	V_{SS}	B8	$\overline{XINT7}$	A11

NOTE: Do not connect any external logic to pins marked NC (no connect pins).

A6	V_{CC}	C11	$\overline{XINT0}$	H14	CLKIN	N1	AD27
A7	V_{CC}	C12	$\overline{TRST}$	J1	V_{CC}	N2	AD26
A8	V_{CC}	C13	STEST	J2	V_{SS}	N3	AD24
A9	V_{CC}	C14	NC	J3	$\overline{BE1}$	N4	AD20
A10	$\overline{NMI}$	D1	V_{CC}	J12	NC	N5	V_{SS}
A11	$\overline{XINT7}$	D2	V_{SS}	J13	V_{SS}	N6	V_{SS}
A12	$\overline{XINT5}$	D3	DT/$\overline{R}$	J14	V_{CC}	N7	V_{SS}
A13	$\overline{XINT2}$	D12	TDI	K1	V_{CC}	N8	V_{SS}
A14	TMS	D13	V_{SS}	K2	V_{SS}	N9	V_{SS}
B1	W/$\overline{R}$	D14	V_{CC}	K3	AD31	N10	V_{SS}
B2	D/$\overline{C}$	E1	V_{CC}	K12	AD2	N11	V_{SS}
B3	WIDTH/HLTD0	E2	V_{SS}	K13	V_{SS}	N12	AD10
B4	TDO	E3	$\overline{DEN}$	K14	V_{CC}	N13	AD7
B5	NC	E12	$\overline{RESET}$	L1	$\overline{BE2}$	N14	AD3
B6	V_{SS}	E13	V_{SS}	L2	$\overline{BE3}$	P1	AD25
B7	V_{SS}	E14	V_{CC}	L3	AD28	P2	AD22
B8	V_{SS}	F1	V_{CC}	L12	AD5	P3	AD19
B9	V_{SS}	F2	V_{SS}	L13	AD1	P4	AD18
B10	$\overline{XINT6}$	F3	BSTAT	L14	V_{CC}	P5	V_{CC}
B11	$\overline{XINT4}$	F12	$\overline{RDYRCV}$	M1	AD30	P6	V_{CC}
B12	$\overline{XINT3}$	F13	V_{SS}	M2	AD29	P7	V_{CC}
B13	TCK	F14	V_{CC}	M3	NC	P8	V_{CC}
B14	NC	G1	V_{CC}	M4	AD23	P9	V_{CC}
C1	$\overline{LOCK/ONCE}$	G2	V_{SS}	M5	AD21	P10	V_{CC}
C2	HOLDA	G3	ALE	M6	AD17	P11	V_{CC}
C3	$\overline{BLAST}$	G12	NC	M7	AD16	P12	AD13
C4	A3	G13	V_{SS}	M8	AD15	P13	AD11
C5	A2	G14	V_{CC}	M9	AD14	P14	AD6

NOTE: Do not connect any external logic to pins marked NC (no connect pins).

3.1.3 80960Jx PQFP Pinout

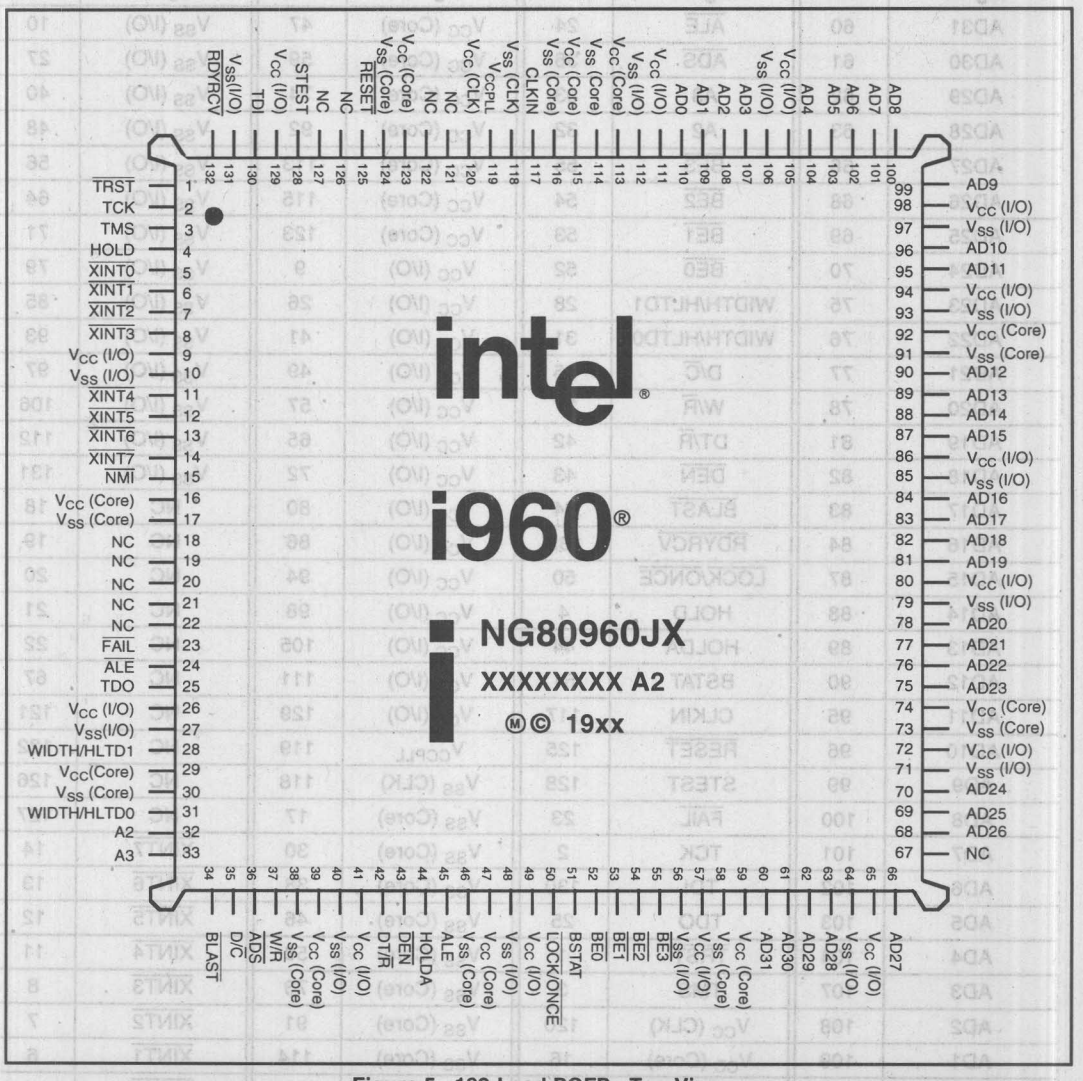


Figure 5. 132-Lead PQFP - Top View

Table 8. 132-Lead PQFP Pinout — In Signal Order

Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin
AD31	60	$\overline{ALE}$	24	V_{CC} (Core)	47	V_{SS} (I/O)	10
AD30	61	$\overline{ADS}$	36	V_{CC} (Core)	59	V_{SS} (I/O)	27
AD29	62	A3	33	V_{CC} (Core)	74	V_{SS} (I/O)	40
AD28	63	A2	32	V_{CC} (Core)	92	V_{SS} (I/O)	48
AD27	66	$\overline{BE3}$	55	V_{CC} (Core)	113	V_{SS} (I/O)	56
AD26	68	$\overline{BE2}$	54	V_{CC} (Core)	115	V_{SS} (I/O)	64
AD25	69	$\overline{BE1}$	53	V_{CC} (Core)	123	V_{SS} (I/O)	71
AD24	70	$\overline{BE0}$	52	V_{CC} (I/O)	9	V_{SS} (I/O)	79
AD23	75	WIDTH/HLTD1	28	V_{CC} (I/O)	26	V_{SS} (I/O)	85
AD22	76	WIDTH/HLTD0	31	V_{CC} (I/O)	41	V_{SS} (I/O)	93
AD21	77	D/$\overline{C}$	35	V_{CC} (I/O)	49	V_{SS} (I/O)	97
AD20	78	W/$\overline{R}$	37	V_{CC} (I/O)	57	V_{SS} (I/O)	106
AD19	81	DT/$\overline{R}$	42	V_{CC} (I/O)	65	V_{SS} (I/O)	112
AD18	82	$\overline{DEN}$	43	V_{CC} (I/O)	72	V_{SS} (I/O)	131
AD17	83	$\overline{BLAST}$	34	V_{CC} (I/O)	80	NC	18
AD16	84	$\overline{RDYRCV}$	132	V_{CC} (I/O)	86	NC	19
AD15	87	$\overline{LOCK/ONCE}$	50	V_{CC} (I/O)	94	NC	20
AD14	88	HOLD	4	V_{CC} (I/O)	98	NC	21
AD13	89	HOLDA	44	V_{CC} (I/O)	105	NC	22
AD12	90	BSTAT	51	V_{CC} (I/O)	111	NC	67
AD11	95	CLKIN	117	V_{CC} (I/O)	129	NC	121
AD10	96	$\overline{RESET}$	125	V_{CCPLL}	119	NC	122
AD9	99	STEST	128	V_{SS} (CLK)	118	NC	126
AD8	100	$\overline{FAIL}$	23	V_{SS} (Core)	17	NC	127
AD7	101	TCK	2	V_{SS} (Core)	30	$\overline{XINT7}$	14
AD6	102	TDI	130	V_{SS} (Core)	38	$\overline{XINT6}$	13
AD5	103	TDO	25	V_{SS} (Core)	46	$\overline{XINT5}$	12
AD4	104	$\overline{TRST}$	1	V_{SS} (Core)	58	$\overline{XINT4}$	11
AD3	107	TMS	3	V_{SS} (Core)	73	$\overline{XINT3}$	8
AD2	108	V_{CC} (CLK)	120	V_{SS} (Core)	91	$\overline{XINT2}$	7
AD1	109	V_{CC} (Core)	16	V_{SS} (Core)	114	$\overline{XINT1}$	6
AD0	110	V_{CC} (Core)	29	V_{SS} (Core)	116	$\overline{XINT0}$	5
ALE	45	V_{CC} (Core)	39	V_{SS} (Core)	124	$\overline{NMI}$	15

NOTE: Do not connect any external logic to pins marked NC (no connect pins).

Table 9. 132-Lead PQFP Pinout — In Pin Order

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
1	$\overline{TRST}$	34	$\overline{BLAST}$	67	NC	100	AD8
2	TCK	35	D/$\overline{C}$	68	AD26	101	AD7
3	TMS	36	$\overline{ADS}$	69	AD25	102	AD6
4	HOLD	37	W/$\overline{R}$	70	AD24	103	AD5
5	$\overline{XINT0}$	38	V_{SS} (Core)	71	V_{SS} (I/O)	104	AD4
6	$\overline{XINT1}$	39	V_{CC} (Core)	72	V_{CC} (I/O)	105	V_{CC} (I/O)
7	$\overline{XINT2}$	40	V_{SS} (I/O)	73	V_{SS} (Core)	106	V_{SS} (I/O)
8	$\overline{XINT3}$	41	V_{CC} (I/O)	74	V_{CC} (Core)	107	AD3
9	V_{CC} (I/O)	42	DT/$\overline{R}$	75	AD23	108	AD2
10	V_{SS} (I/O)	43	$\overline{DEN}$	76	AD22	109	AD1
11	$\overline{XINT4}$	44	HOLDA	77	AD21	110	AD0
12	$\overline{XINT5}$	45	ALE	78	AD20	111	V_{CC} (I/O)
13	$\overline{XINT6}$	46	V_{SS} (Core)	79	V_{SS} (I/O)	112	V_{SS} (I/O)
14	$\overline{XINT7}$	47	V_{CC} (Core)	80	V_{CC} (I/O)	113	V_{CC} (Core)
15	$\overline{NMI}$	48	V_{SS} (I/O)	81	AD19	114	V_{SS} (Core)
16	V_{CC} (Core)	49	V_{CC} (I/O)	82	AD18	115	V_{CC} (Core)
17	V_{SS} (Core)	50	$\overline{LOCK/ONCE}$	83	AD17	116	V_{SS} (Core)
18	NC	51	BSTAT	84	AD16	117	CLKIN
19	NC	52	$\overline{BE0}$	85	V_{SS} (I/O)	118	V_{SS} (CLK)
20	NC	53	$\overline{BE1}$	86	V_{CC} (I/O)	119	V_{CCPLL}
21	NC	54	$\overline{BE2}$	87	AD15	120	V_{CC} (CLK)
22	NC	55	$\overline{BE3}$	88	AD14	121	NC
23	$\overline{FAIL}$	56	V_{SS} (I/O)	89	AD13	122	NC
24	$\overline{ALE}$	57	V_{CC} (I/O)	90	AD12	123	V_{CC} (Core)
25	TDO	58	V_{SS} (Core)	91	V_{SS} (Core)	124	V_{SS} (Core)
26	V_{CC} (I/O)	59	V_{CC} (Core)	92	V_{CC} (Core)	125	$\overline{RESET}$
27	V_{SS} (I/O)	60	AD31	93	V_{SS} (I/O)	126	NC
28	WIDTH/HLTD1	61	AD30	94	V_{CC} (I/O)	127	NC
29	V_{CC} (Core)	62	AD29	95	AD11	128	STEST
30	V_{SS} (Core)	63	AD28	96	AD10	129	V_{CC} (I/O)
31	WIDTH/HLTD0	64	V_{SS} (I/O)	97	V_{SS} (I/O)	130	TDI
32	A2	65	V_{CC} (I/O)	98	V_{CC} (I/O)	131	V_{SS} (I/O)
33	A3	66	AD27	99	AD9	132	$\overline{RDYRCV}$

NOTE: Do not connect any external logic to pins marked NC (no connect pins).

3.2 Package Thermal Specifications

The 80960JD is specified for operation when T_C (case temperature) is within the range of 0°C to 85°C for the (PGA) 80960JD-50, or 0°C to 100°C for the (PQFP and PGA) 80960JD-40 and 80960JD-33. Case temperature may be measured in any environment to determine whether the 80960JD is within specified operating range. The case temperature should be measured at the center of the top surface, opposite the pins.

θ_{CA} is the thermal resistance from case to ambient. Use the following equation to calculate T_A, the maximum ambient temperature to conform to a particular case temperature:

$$T_A = T_C - P\ (\theta_{CA})$$

Junction temperature (T_J) is commonly used in reliability calculations. T_J can be calculated from θ_{JC} (thermal resistance from junction to case) using the following equation:

$$T_J = T_C + P\ (\theta_{JC})$$

Similarly, if T_A is known, the corresponding case temperature (T_C) can be calculated as follows:

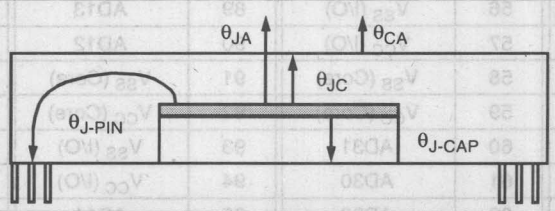
$$T_C = T_A + P\ (\theta_{CA})$$

Compute P by multiplying I_{CC} from Table 14 and V_{CC}. Values for θ_{JC} and θ_{CA} are given in Table 10 for the PGA package and Table 11 for the PQFP package. For high speed operation, the processor's θ_{JA} may be significantly reduced by adding a heatsink and/or by increasing airflow.

Figure 6 shows the maximum ambient temperature (T_A) permitted without exceeding T_C for the 80960JD-50 in a PGA package. Figure 7 illustrates this for the 80960JD-40 in PGA and PQFP packages. The curves are based on minimum I_{CC} (hot) and maximum V_{CC} of +5.25 V, with a T_{CASE} of +85°C for the 80960JD-50, or a T_{CASE} of +100°C for the 80960JD-40.

Table 10. 132-Lead PGA Package Thermal Characteristics

Parameter	Thermal Resistance — °C/Watt					
	Airflow — ft./min (m/sec)					
	0 (0)	200 (1.01)	400 (2.03)	600 (3.04)	800 (4.06)	1000 (5.08)
θ_{JC} (Junction-to-Case)	3	3	3	3	3	3
θ_{CA} (Case-to-Ambient) (No Heatsink)	18	15	12	11	11	11
θ_{CA} (Case-to-Ambient) (Omnidirectional Heatsink)	15	12	9	8	8	8
θ_{CA} (Case-to-Ambient) (Unidirectional Heatsink)	14	11	8	7	7	7

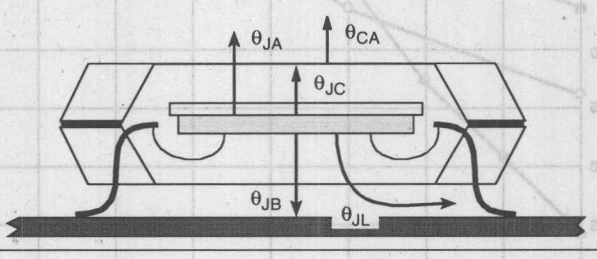


NOTES:

1. This table applies to a PGA device plugged into a socket or soldered directly into a board.
2. $\theta_{JA} = \theta_{JC} + \theta_{CA}$
3. $\theta_{J\text{-}CAP}$ = 4°C/W (approx.)
4. $\theta_{J\text{-}PIN}$ = 4°C/W (inner pins) (approx.)
5. $\theta_{J\text{-}PIN}$ = 8°C/W (outer pins) (approx.)

Table 11. 132-Lead PQFP Package Thermal Characteristics

Thermal Resistance — °C/Watt							
Parameter	Airflow — ft./min (m/sec)						
	0 (0)	50 (0.25)	100 (0.50)	200 (1.01)	400 (2.03)	600 (3.04)	800 (4.06)
θ_{JC} (Junction-to-Case)	6	7	7	7	7	7	7
θ_{CA} (Case-to-Ambient -No Heatsink)	23	20	18	14	10	9	8



NOTES:

1. This table applies to a PQFP device soldered directly into board.
2. $\theta_{JA} = \theta_{JC} + \theta_{CA}$
3. θ_{JL} = 18°C/W (approx.)
4. θ_{JB} = 18°C/W (approx.)

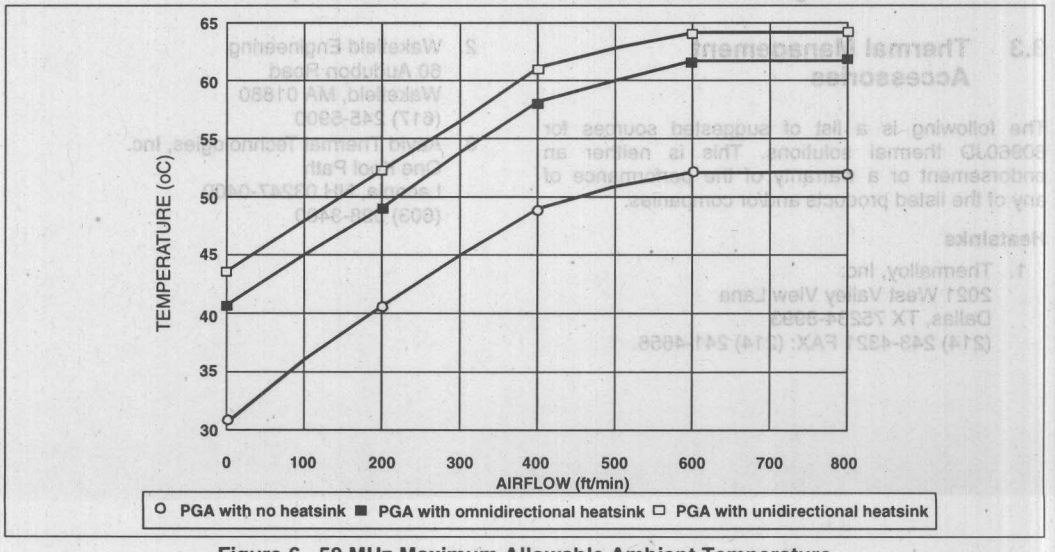


Figure 6. 50 MHz Maximum Allowable Ambient Temperature

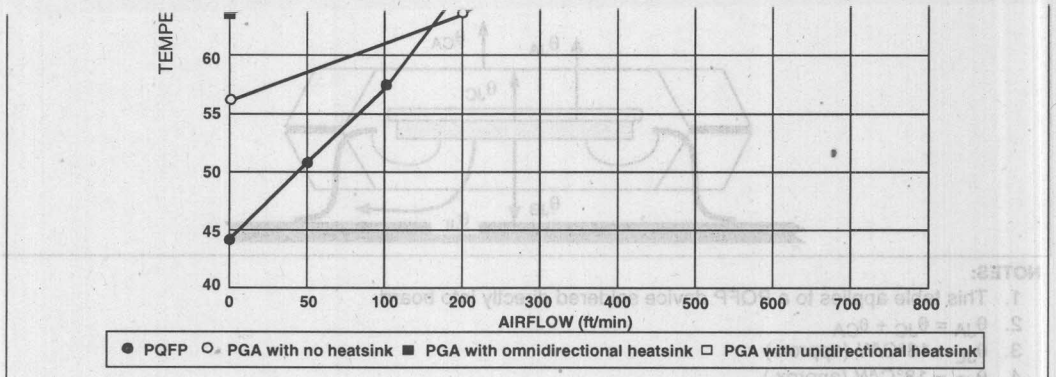


Figure 7. 40 MHz Maximum Allowable Ambient Temperature

3.3 Thermal Management Accessories

The following is a list of suggested sources for 80960JD thermal solutions. This is neither an endorsement or a warranty of the performance of any of the listed products and/or companies.

Heatsinks

1. Thermalloy, Inc.
   2021 West Valley View Lane
   Dallas, TX 75234-8993
   (214) 243-4321 FAX: (214) 241-4656
2. Wakefield Engineering
   60 Audubon Road
   Wakefield, MA 01880
   (617) 245-5900
3. Aavid Thermal Technologies, Inc.
   One Kool Path
   Laconia, NH 03247-0400
   (603) 528-3400

4.0 ELECTRICAL SPECIFICATIONS

4.1 Absolute Maximum Ratings

Parameter	Maximum Rating
Storage Temperature	–65° C to +150° C
Case Temperature Under Bias	–65° C to +110° C
Supply Voltage wrt. V_{SS}	–0.5V to + 4.6V
Voltage on Other Pins wrt. V_{SS}	–0.5V to V_{CC} + 0.5V

NOTICE: This data sheet contains preliminary information on new products in production. The specifications are subject to change without notice.

WARNING: *Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

4.2 Operating Conditions

Table 12. 80960JD Operating Conditions

Symbol	Parameter	Min	Max	Units	Notes
V_{CC}	Supply Voltage				
	80960JD-50	4.75	5.25	V	
	80960JD-40	4.75	5.25		
	80960JD-33	4.75	5.25		
f_{CLKIN}	Input Clock Frequency				
	80960JD-50	8	25	MHz	
	80960JD-40	8	20		
	80960JD-33	8	16.67		
T_C	Operating Case Temperature				
	A80960JD-50 (132 PGA)	0	85	°C	
	A80960JD-40 (132 PGA)	0	100		
	A80960JD-33 (132 PGA)	0	100		
	NG80960JD-40 (132 PQFP)	0	100		
	NG80960JD-33 (132 PQFP)	0	100		

4.3 Connection Recommendations

For clean on-chip power distribution, V_{CC} and V_{SS} pins separately feed the device's functional units. Power and ground connections must be made to all 80960JD power and ground pins. On the circuit board, every V_{CC} pin should connect to a power plane and every V_{SS} pin should connect to a ground plane. Place liberal decoupling capacitance near the 80960JD, since the processor can cause transient power surges.

Pay special attention to the Test Reset ($\overline{TRST}$) pin. It is essential that the JTAG Boundary Scan Test Access Port (TAP) controller initializes to a known state whether it will be used or not. If the JTAG Boundary Scan function will be used, connect a pulldown resistor between the $\overline{TRST}$ pin and V_{SS}. If the JTAG Boundary Scan function will not be used (even for board-level testing), connect the $\overline{TRST}$ pin to V_{SS}. Also, do not connect the TDI, TDO, and TCK pins if the TAP Controller will not be used.

Pins identified as NC must not be connected in the system.

4.4 DC Specifications

Table 13. 80960JD DC Characteristics

Symbol	Parameter	Min	Typ	Max	Units	Notes
V_{IL}	Input Low Voltage	-0.3		0.8	V	
V_{IH}	Input High Voltage	2.0		V_{CC} + 0.3	V	
V_{OL}	Output Low Voltage			0.45	V	I_{OL} = 5 mA
V_{OH}	Output High Voltage	2.4 V_{CC} - 0.5			V	I_{OH} = -1 mA I_{OH} = -200 µA
V_{OLP}	Output Ground Bounce		< 0.8		V	(1,2)
C_{IN}	Input Capacitance PGA PQFP			 12 10	 pF	$f_{CLKIN} = f_{MIN}$ (2)
C_{OUT}	I/O or Output Capacitance PGA PQFP			 12 10	 pF	$f_{CLKIN} = f_{MIN}$ (2)
C_{CLK}	CLKIN Capacitance PGA PQFP			 12 10	 pF	$f_{CLKIN} = f_{MIN}$ (2)

NOTES:

1. Typical is measured with V_{CC} = 5.0V and temperature = 25 °C.
2. Not tested.

Table 14. 80960JD I_{CC} Characteristics

Symbol	Parameter	Typ	Max	Units	Notes
I_{LI1}	Input Leakage Current for each pin except TCK, TDI, $\overline{TRST}$ and TMS		± 1	µA	$0 \le V_{IN} \le V_{CC}$
I_{LI2}	Input Leakage Current for TCK, TDI, $\overline{TRST}$ and TMS	-140	-250	µA	V_{IN} = 0.45V (1)
I_{LO}	Output Leakage Current		± 1	µA	$0.4 \le V_{OUT} \le V_{CC}$
I_{CC} Active (Power Supply)	80960JD-50		640	mA	(2,3)
	80960JD-40		530		(2,3)
	80960JD-33		450		(2,3)
I_{CC} Active (Thermal)	80960JD-50	525		mA	(2,4)
	80960JD-40	430			(2,4)
	80960JD-33	365			(2,4)
I_{CC} Test (Power modes)	Reset mode				
	80960JD-50		510	mA	(5)
	80960JD-40		430		(5)
	80960JD-33		370		(5)
	Halt mode				
	80960JD-50		48		(5)
	80960JD-40		41		(5)
	80960JD-33		36		(5)
	ONCE mode		10		(5)

NOTES:

1. These pins have internal pullup devices. Typical leakage current is not tested.
2. Measured with device operating and outputs loaded to the test condition in Figure 8, AC Test Load (pg. 33).
3. I_{CC} Active (Power Supply) value is provided for selecting your system's power supply. It is measured using one of the worst case instruction mixes with V_{CC} = 5.25V. This parameter is characterized but not tested.
4. I_{CC} Active (Thermal) value is provided for your system's thermal management. Typical I_{CC} is measured with V_{CC} = 5.0V and temperature = 25° C. This parameter is characterized but not tested.
5. I_{CC} Test (Power modes) refers to the I_{CC} values that are tested when the 80960JD is in Reset mode, Halt mode or ONCE mode with V_{CC} = 5.25V.

4.5 AC Specifications

The 80960JD AC timings are based upon device characterization.

Table 15. 80960JD AC Characteristics (50 MHz) (Sheet 1 of 2)

Symbol	Parameter	Min	Max	Units	Notes
	INPUT CLOCK TIMINGS				
T_F	CLKIN Frequency	8	25	MHz	
T_C	CLKIN Period	40	125	ns	
T_{CS}	CLKIN Period Stability		± 250	ps	(1, 2)
T_{CH}	CLKIN High Time	16		ns	Measured at 1.5 V (1)
T_{CL}	CLKIN Low Time	16		ns	Measured at 1.5 V (1)
T_{CR}	CLKIN Rise Time		25	ns	0.8 V to 2.0 V (1)
T_{CF}	CLKIN Fall Time		5	ns	2.0 V to 0.8 V (1)
	SYNCHRONOUS OUTPUT TIMINGS				
T_{OV1}	Output Valid Delay, Except ALE/$\overline{ALE}$ Inactive and DT/$\overline{R}$	3.5	17	ns	(3)
T_{OV2}	Output Valid Delay, DT/$\overline{R}$	$0.5T_C + 3.5$	$0.5T_C + 17$	ns	
T_{OF}	Output Float Delay	3.5	15	ns	(4)
	SYNCHRONOUS INPUT TIMINGS				
T_{IS1}	Input Setup to CLKIN — AD31:0, $\overline{NMI}$, $\overline{XINT7:0}$	8		ns	(5)
T_{IH1}	Input Hold from CLKIN — AD31:0, $\overline{NMI}$, $\overline{XINT7:0}$	2		ns	(5)
T_{IS2}	Input Setup to CLKIN — $\overline{RDYRCV}$ and HOLD	9		ns	(6)
T_{IH2}	Input Hold from CLKIN — $\overline{RDYRCV}$ and HOLD	1		ns	(6)
T_{IS3}	Input Setup to CLKIN — $\overline{RESET}$	8		ns	(7)
T_{IH3}	Input Hold from CLKIN — $\overline{RESET}$	2		ns	(7)
T_{IS4}	Input Setup to $\overline{RESET}$ — $\overline{ONCE}$, STEST	8		ns	(8)
T_{IH4}	Input Hold from $\overline{RESET}$ — $\overline{ONCE}$, STEST	2		ns	(8)

NOTE: See Table 16 on page 28 for note definitions for this table.

T_{DXD}	DT/R Valid to DEN Active				Equal Loading (9)
BOUNDARY SCAN TEST SIGNAL TIMINGS					
T_{BSF}	TCK Frequency		$0.5T_F$	MHz	
T_{BSCH}	TCK High Time	15		ns	Measured at 1.5 V (1)
T_{BSCL}	TCK Low Time	15		ns	Measured at 1.5 V (1)
T_{BSCR}	TCK Rise Time		5	ns	0.8 V to 2.0 V (1)
T_{BSCF}	TCK Fall Time		5	ns	2.0 V to 0.8 V (1)
T_{BSIS1}	Input Setup to TCK — TDI, TMS	4		ns	
T_{BSIH1}	Input Hold from TCK — TDI, TMS	6		ns	
T_{BSOV1}	TDO Valid Delay	3	30	ns	(1,10)
T_{BSOF1}	TDO Float Delay	3	30	ns	(1,10)
T_{BSOV2}	All Outputs (Non-Test) Valid Delay	3	30	ns	(1,10)
T_{BSOF2}	All Outputs (Non-Test) Float Delay	3	30	ns	(1,10)
T_{BSIS2}	Input Setup to TCK — All Inputs (Non-Test)	4		ns	
T_{BSIH2}	Input Hold from TCK — All Inputs (Non-Test)	6		ns	

NOTE: See Table 16 on page 28 for note definitions for this table.

Table 16. Note Definitions for Table 15, 80960JD AC Characteristics (50 MHz) (pg. 26)

NOTES:

1. Not tested.
2. To ensure a 1:1 relationship between the amplitude of the input jitter and the internal clock, the jitter frequency spectrum should not have any power peaking between 500 KHz and 1/3 of the CLKIN frequency.
3. Inactive ALE/$\overline{\text{ALE}}$ refers to the falling edge of ALE and the rising edge of $\overline{\text{ALE}}$. For inactive ALE/$\overline{\text{ALE}}$ timings, refer to Relative Output Timings in this table.
4. A float condition occurs when the output current becomes less than I_{LO}. Float delay is not tested, but is designed to be no longer than the valid delay.
5. AD31:0 are synchronous inputs. Setup and hold times must be met for proper processor operation. $\overline{\text{NMI}}$ and $\overline{\text{XINT7:0}}$ may be synchronous or asynchronous. Meeting setup and hold time guarantees recognition at a particular clock edge. For asynchronous operation, $\overline{\text{NMI}}$ and $\overline{\text{XINT7:0}}$ must be asserted for a minimum of two CLKIN periods to guarantee recognition.
6. $\overline{\text{RDYRCV}}$ and HOLD are synchronous inputs. Setup and hold times must be met for proper processor operation.
7. $\overline{\text{RESET}}$ may be synchronous or asynchronous. Meeting setup and hold time guarantees recognition at a particular clock edge.
8. $\overline{\text{ONCE}}$ and $\overline{\text{STEST}}$ must be stable at the rising edge of $\overline{\text{RESET}}$ for proper operation.
9. Guaranteed by design. May not be 100% tested.
10. Relative to falling edge of TCK.

Table 17. 80960JD AC Characteristics (40 MHz) (Sheet 1 of 3)

Symbol	Parameter	Min	Max	Units	Notes
	INPUT CLOCK TIMINGS				
T_F	CLKIN Frequency	8	20	MHz	
T_C	CLKIN Period	50	125	ns	
T_{CS}	CLKIN Period Stability		±250	ps	(1, 2)
T_{CH}	CLKIN High Time	20		ns	Measured at 1.5 V (1)
T_{CL}	CLKIN Low Time	20		ns	Measured at 1.5 V (1)
T_{CR}	CLKIN Rise Time		7	ns	0.8 V to 2.0 V (1)
T_{CF}	CLKIN Fall Time		7	ns	2.0 V to 0.8 V (1)
	SYNCHRONOUS OUTPUT TIMINGS				
T_{OV1}	Output Valid Delay, Except ALE/$\overline{\text{ALE}}$ Inactive and DT/$\overline{\text{R}}$	3.5	18	ns	(3)

Table 17. 80960JD AC Characteristics (40 MHz) (Sheet 2 of 3)

Symbol	Parameter	Min	Max	Units	Notes
T_{OV2}	Output Valid Delay, DT/$\overline{R}$	$0.5T_C + 3.5$	$0.5T_C + 18$	ns	
T_{OF}	Output Float Delay	3.5	16	ns	(4)
SYNCHRONOUS INPUT TIMINGS					
T_{IS1}	Input Setup to CLKIN — AD31:0, $\overline{NMI}$, $\overline{XINT7:0}$	8		ns	(5)
T_{IH1}	Input Hold from CLKIN — AD31:0, $\overline{NMI}$, $\overline{XINT7:0}$	2		ns	(5)
T_{IS2}	Input Setup to CLKIN — $\overline{RDYRCV}$ and HOLD	9		ns	(6)
T_{IH2}	Input Hold from CLKIN — $\overline{RDYRCV}$ and HOLD	1		ns	(6)
T_{IS3}	Input Setup to CLKIN — $\overline{RESET}$	8		ns	(7)
T_{IH3}	Input Hold from CLKIN — $\overline{RESET}$	2		ns	(7)
T_{IS4}	Input Setup to $\overline{RESET}$ — $\overline{ONCE}$, STEST	8		ns	(8)
T_{IH4}	Input Hold from $\overline{RESET}$ — $\overline{ONCE}$, STEST	2		ns	(8)
RELATIVE OUTPUT TIMINGS					
T_{LXL}	ALE/$\overline{ALE}$ Width	$0.5T_C - 7.5$		ns	(9)
T_{LXA}	Address Hold from ALE/$\overline{ALE}$ Inactive				Equal Loading (9)
T_{DXD}	DT/$\overline{R}$ Valid to $\overline{DEN}$ Active				Equal Loading (9)
BOUNDARY SCAN TEST SIGNAL TIMINGS					
T_{BSF}	TCK Frequency		$0.5T_F$	MHz	
T_{BSCH}	TCK High Time	15		ns	Measured at 1.5 V (1)
T_{BSCL}	TCK Low Time	15		ns	Measured at 1.5 V (1)
T_{BSCR}	TCK Rise Time		5	ns	0.8 V to 2.0 V (1)
T_{BSCF}	TCK Fall Time		5	ns	2.0 V to 0.8 V (1)
T_{BSIS1}	Input Setup to TCK — TDI, TMS	4		ns	
T_{BSIH1}	Input Hold from TCK — TDI, TMS	6		ns	
T_{BSOV1}	TDO Valid Delay	3	30	ns	(1, 10)
T_{BSOF1}	TDO Float Delay	3	30	ns	(1, 10)

Table 17. 80960JD AC Characteristics (40 MHz) (Sheet 3 of 3)

Symbol	Parameter	Min	Max	Units	Notes
T_{BSOV2}	All Outputs (Non-Test) Valid Delay	3	30	ns	(1, 10)
T_{BSOF2}	All Outputs (Non-Test) Float Delay	3	30	ns	(1, 10)
T_{BSIS2}	Input Setup to TCK — All Inputs (Non-Test)	4		ns	
T_{BSIH2}	Input Hold from TCK — All Inputs (Non-Test)	6		ns	

NOTES:

1. Not tested.
2. To ensure a 1:1 relationship between the amplitude of the input jitter and the internal clock, the jitter frequency spectrum should not have any power peaking between 500 KHz and 1/3 of the CLKIN frequency.
3. Inactive ALE/$\overline{\text{ALE}}$ refers to the falling edge of ALE and the rising edge of $\overline{\text{ALE}}$. For inactive ALE/$\overline{\text{ALE}}$ timings, refer to Relative Output Timings in this table.
4. A float condition occurs when the output current becomes less than I_{LO}. Float delay is not tested, but is designed to be no longer than the valid delay.
5. AD31:0 are synchronous inputs. Setup and hold times must be met for proper processor operation. $\overline{\text{NMI}}$ and $\overline{\text{XINT7:0}}$ may be synchronous or asynchronous. Meeting setup and hold time guarantees recognition at a particular clock edge. For asynchronous operation, $\overline{\text{NMI}}$ and $\overline{\text{XINT7:0}}$ must be asserted for a minimum of two CLKIN periods to guarantee recognition.
6. $\overline{\text{RDYRCV}}$ and HOLD are synchronous inputs. Setup and hold times must be met for proper processor operation.
7. $\overline{\text{RESET}}$ may be synchronous or asynchronous. Meeting setup and hold time guarantees recognition at a particular clock edge.
8. $\overline{\text{ONCE}}$ and $\overline{\text{STEST}}$ must be stable at the rising edge of $\overline{\text{RESET}}$ for proper operation.
9. Guaranteed by design. May not be 100% tested.
10. Relative to falling edge of TCK.

Table 18. 80960JD AC Characteristics (33 MHz) (Sheet 1 of 2)

Symbol	Parameter	Min	Max	Units	Notes
	INPUT CLOCK TIMINGS				
T_F	CLKIN Frequency	8	16.67	MHz	
T_C	CLKIN Period	60	125	ns	
T_{CS}	CLKIN Period Stability		± 250	ps	(1, 2)
T_{CH}	CLKIN High Time	24		ns	Measured at 1.5 V (1)
T_{CL}	CLKIN Low Time	24		ns	Measured at 1.5 V (1)
T_{CR}	CLKIN Rise Time		8	ns	0.8 V to 2.0 V (1)
T_{CF}	CLKIN Fall Time		8	ns	2.0 V to 0.8 V (1)
	SYNCHRONOUS OUTPUT TIMINGS				
T_{OV1}	Output Valid Delay, Except ALE/$\overline{ALE}$ Inactive and DT/$\overline{R}$	3.5	19	ns	(3)
T_{OV2}	Output Valid Delay, DT/$\overline{R}$	$0.5T_C$ + 3.5	$0.5T_C$ + 19	ns	
T_{OF}	Output Float Delay	3.5	18	ns	(4)
	SYNCHRONOUS INPUT TIMINGS				
T_{IS1}	Input Setup to CLKIN — AD31:0, $\overline{NMI}$, $\overline{XINT7:0}$	8		ns	(5)
T_{IH1}	Input Hold from CLKIN — AD31:0, $\overline{NMI}$, $\overline{XINT7:0}$	2		ns	(5)
T_{IS2}	Input Setup to CLKIN — $\overline{RDYRCV}$ and HOLD	9		ns	(6)
T_{IH2}	Input Hold from CLKIN — $\overline{RDYRCV}$ and HOLD	1		ns	(6)
T_{IS3}	Input Setup to CLKIN — $\overline{RESET}$	8		ns	(7)
T_{IH3}	Input Hold from CLKIN — $\overline{RESET}$	2		ns	(7)
T_{IS4}	Input Setup to $\overline{RESET}$ — $\overline{ONCE}$, STEST	8		ns	(8)
T_{IH4}	Input Hold from $\overline{RESET}$ — $\overline{ONCE}$, STEST	2		ns	(8)
	RELATIVE OUTPUT TIMINGS				
T_{LXL}	ALE/$\overline{ALE}$ Width	$0.5T_C$ - 8		ns	(9)
T_{LXA}	Address Hold from ALE/$\overline{ALE}$ Inactive				Equal Loading (9)
T_{DXD}	DT/$\overline{R}$ Valid to $\overline{DEN}$ Active				Equal Loading (9)

T_{BSCR}	TCK Rise Time		5	ns	0.8 V to 2.0 V (1)
T_{BSCF}	TCK Fall Time		5	ns	2.0 V to 0.8 V (1)
T_{BSIS1}	Input Setup to TCK — TDI, TMS	4		ns	
T_{BSIH1}	Input Hold from TCK — TDI, TMS	6		ns	
T_{BSOV1}	TDO Valid Delay	3	30	ns	(1, 10)
T_{BSOF1}	TDO Float Delay	3	30	ns	(1, 10)
T_{BSOV2}	All Outputs (Non-Test) Valid Delay	3	30	ns	(1, 10)
T_{BSOF2}	All Outputs (Non-Test) Float Delay	3	30	ns	(1, 10)
T_{BSIS2}	Input Setup to TCK — All Inputs (Non-Test)	4		ns	
T_{BSIH2}	Input Hold from TCK — All Inputs (Non-Test)	6		ns	

NOTES:

1. Not tested.
2. To ensure a 1:1 relationship between the amplitude of the input jitter and the internal clock, the jitter frequency spectrum should not have any power peaking between 500 KHz and 1/3 of the CLKIN frequency.
3. Inactive ALE/$\overline{\text{ALE}}$ refers to the falling edge of ALE and the rising edge of $\overline{\text{ALE}}$. For inactive ALE/$\overline{\text{ALE}}$ timings, refer to Relative Output Timings in this table.
4. A float condition occurs when the output current becomes less than I_{LO}. Float delay is not tested, but is designed to be no longer than the valid delay.
5. AD31:0 are synchronous inputs. Setup and hold times must be met for proper processor operation. $\overline{\text{NMI}}$ and $\overline{\text{XINT7:0}}$ may be synchronous or asynchronous. Meeting setup and hold time guarantees recognition at a particular clock edge. For asynchronous operation, $\overline{\text{NMI}}$ and $\overline{\text{XINT7:0}}$ must be asserted for a minimum of two CLKIN periods to guarantee recognition.
6. $\overline{\text{RDYRCV}}$ and HOLD are synchronous inputs. Setup and hold times must be met for proper processor operation.
7. $\overline{\text{RESET}}$ may be synchronous or asynchronous. Meeting setup and hold time guarantees recognition at a particular clock edge.
8. $\overline{\text{ONCE}}$ and $\overline{\text{STEST}}$ must be stable at the rising edge of $\overline{\text{RESET}}$ for proper operation.
9. Guaranteed by design. May not be 100% tested.
10. Relative to falling edge of TCK.

4.5.1 AC Test Conditions and Derating Curves

The AC Specifications in **Section 4.5, AC Specifications** are tested with the 50 pF load indicated in Figure 8. Figure 9 shows how timings vary with load capacitance; Figure 10 shows how output rise and fall times vary with load capacitance.

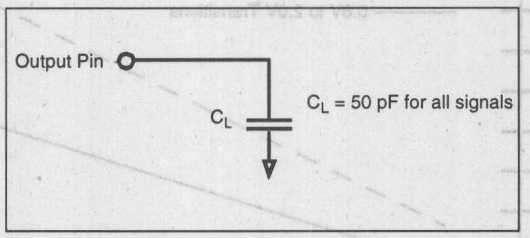


Figure 8. AC Test Load

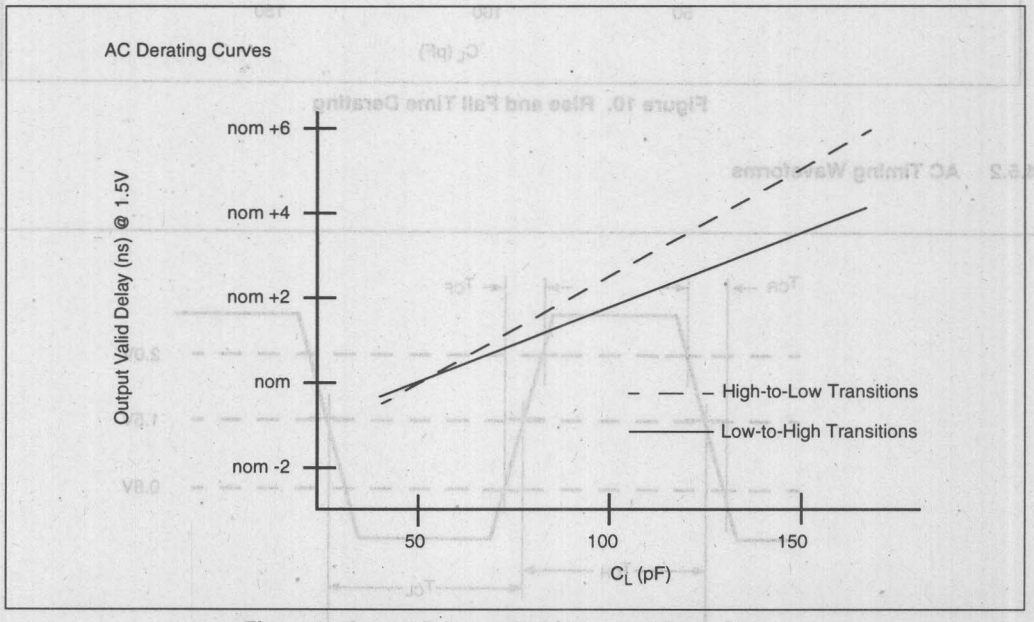


Figure 9. Output Delay or Hold vs. Load Capacitance

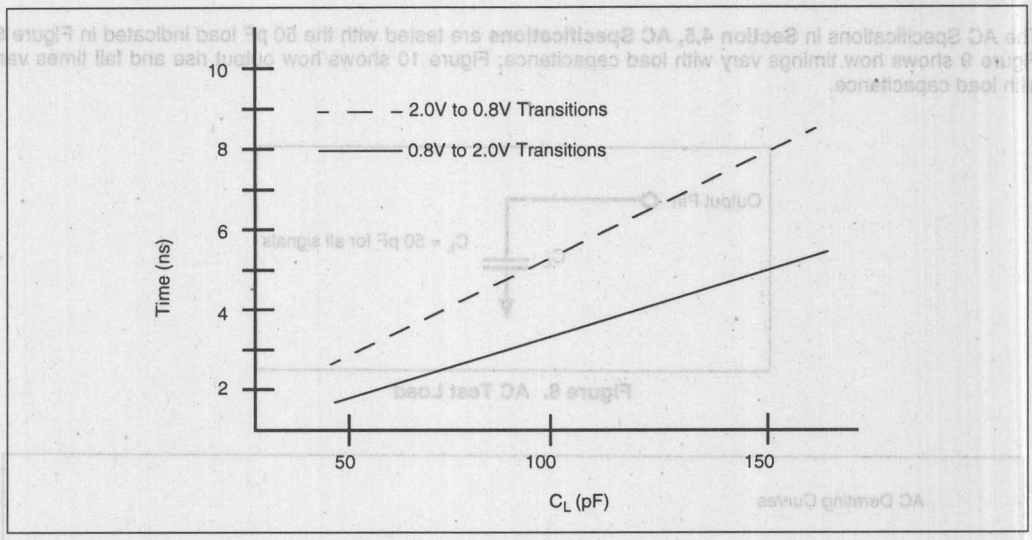


Figure 10. Rise and Fall Time Derating

4.5.2 AC Timing Waveforms

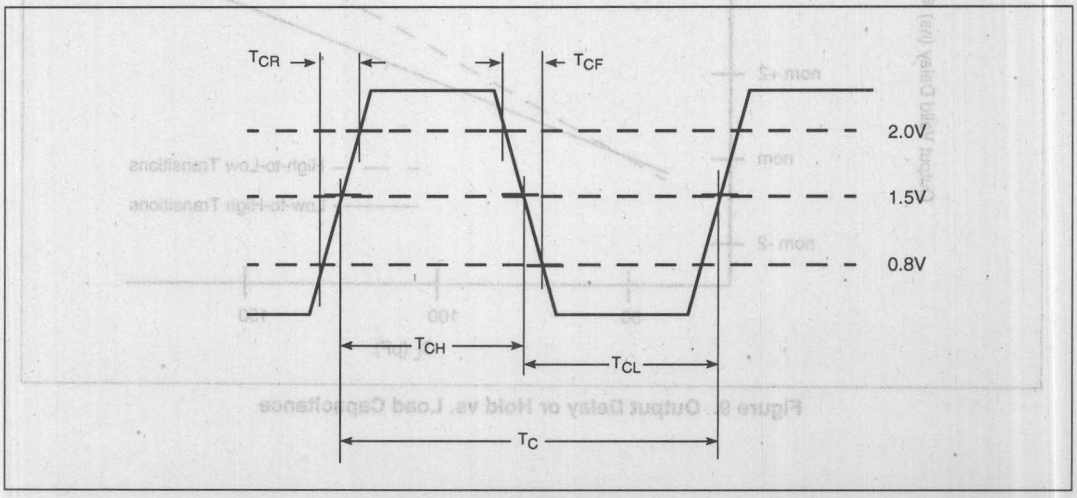


Figure 11. CLKIN Waveform

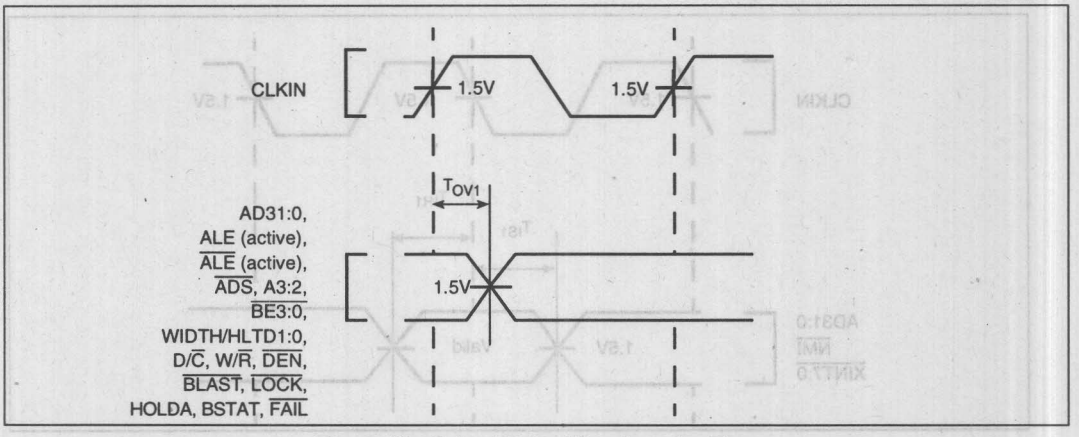


Figure 12. Output Delay Waveform for T_{OV1}

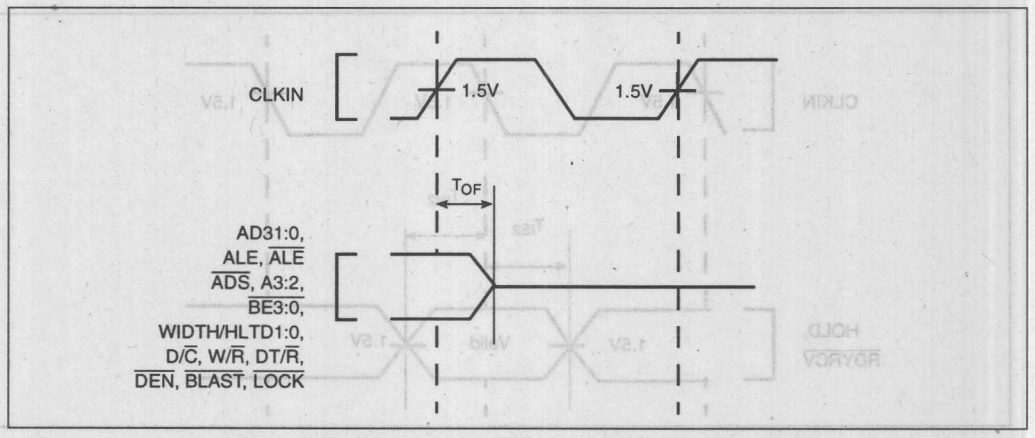


Figure 13. Output Float Waveform for T_{OF}

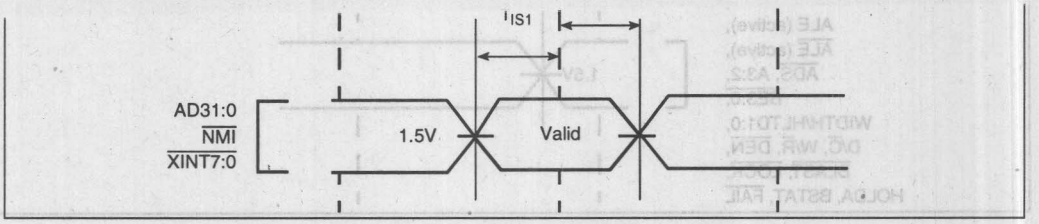


Figure 14. Input Setup and Hold Waveform for T_{IS1} and T_{IH1}

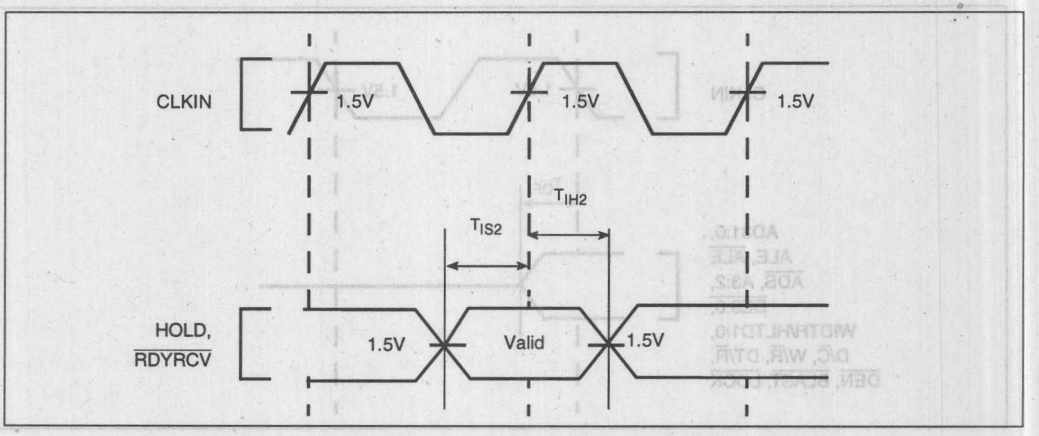


Figure 15. Input Setup and Hold Waveform for T_{IS2} and T_{IH2}

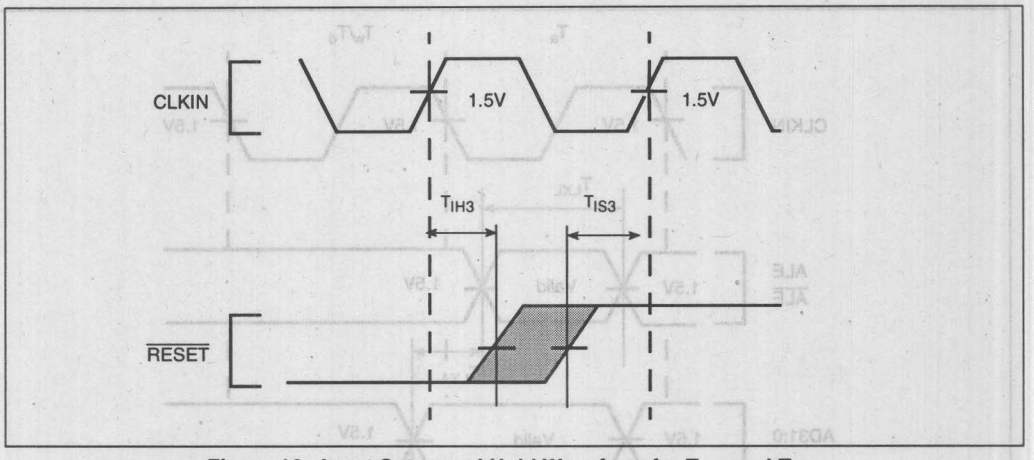


Figure 16. Input Setup and Hold Waveform for T_{IS3} and T_{IH3}

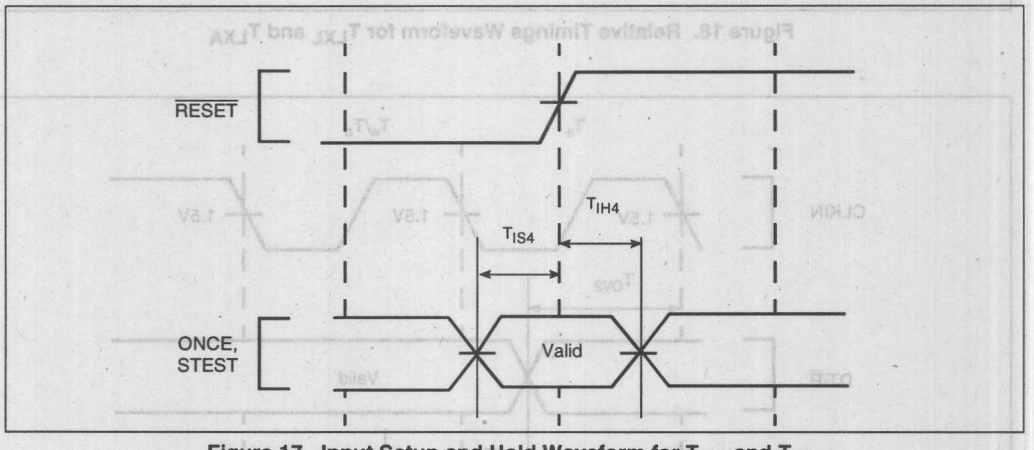


Figure 17. Input Setup and Hold Waveform for T_{IS4} and T_{IH4}

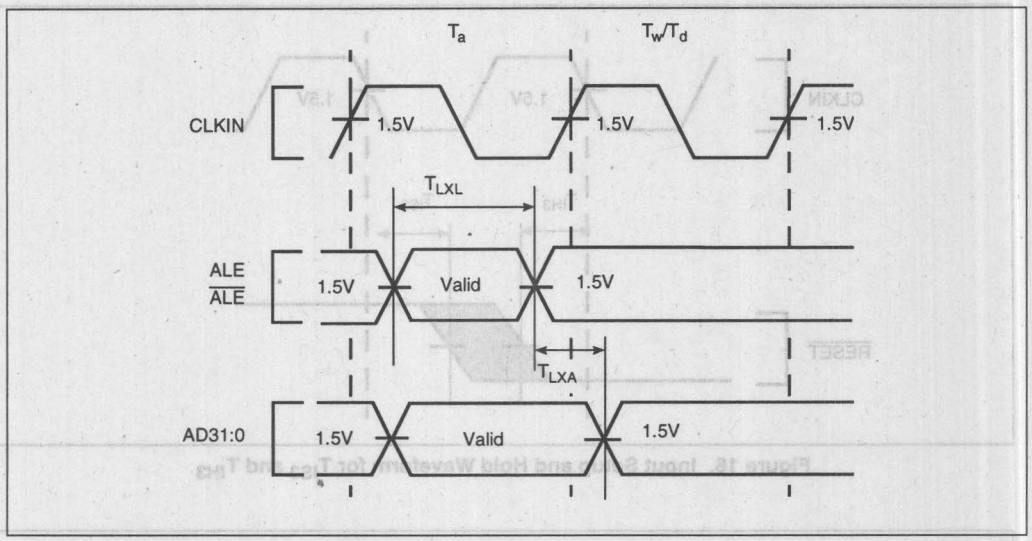


Figure 18. Relative Timings Waveform for T_{LXL} and T_{LXA}

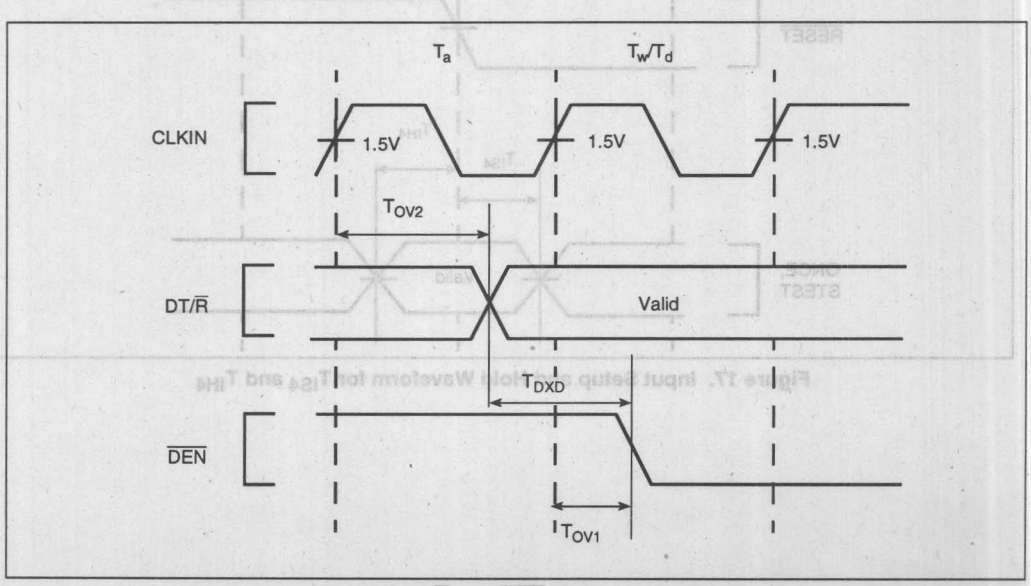


Figure 19. DT/$\overline{R}$ and $\overline{DEN}$ Timings Waveform

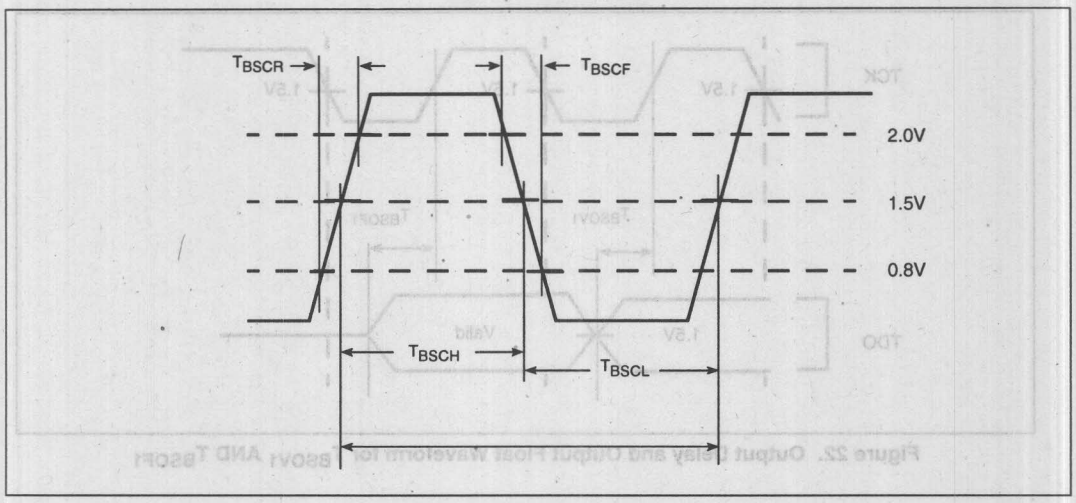


Figure 20. TCK Waveform

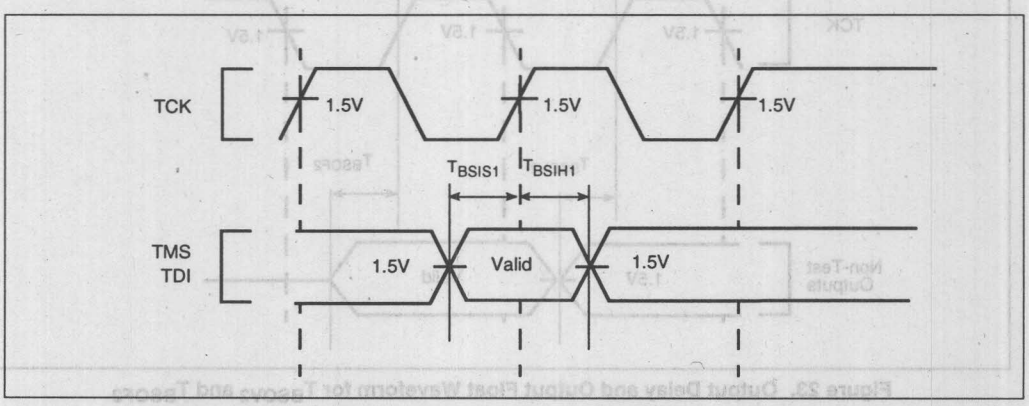


Figure 21. Input Setup and Hold Waveforms for T_{BSIS1} and T_{BSIH1}

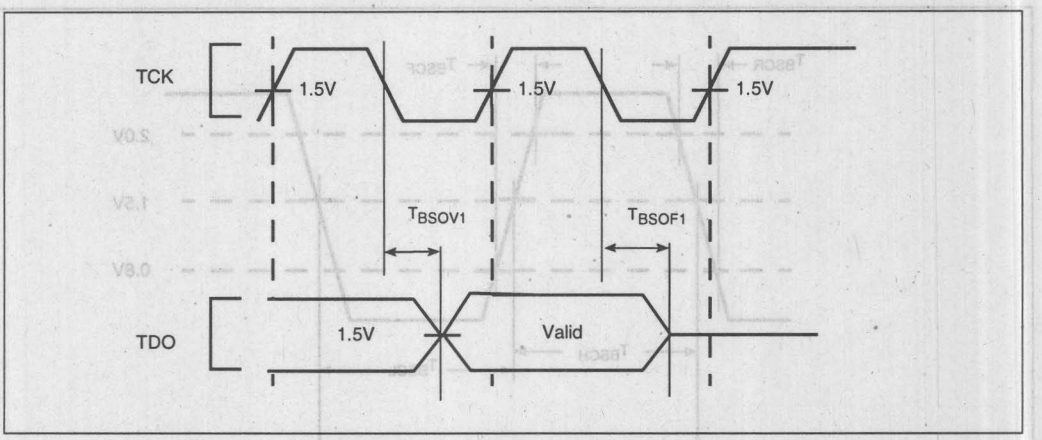


Figure 22. Output Delay and Output Float Waveform for T_{BSOV1} AND T_{BSOF1}

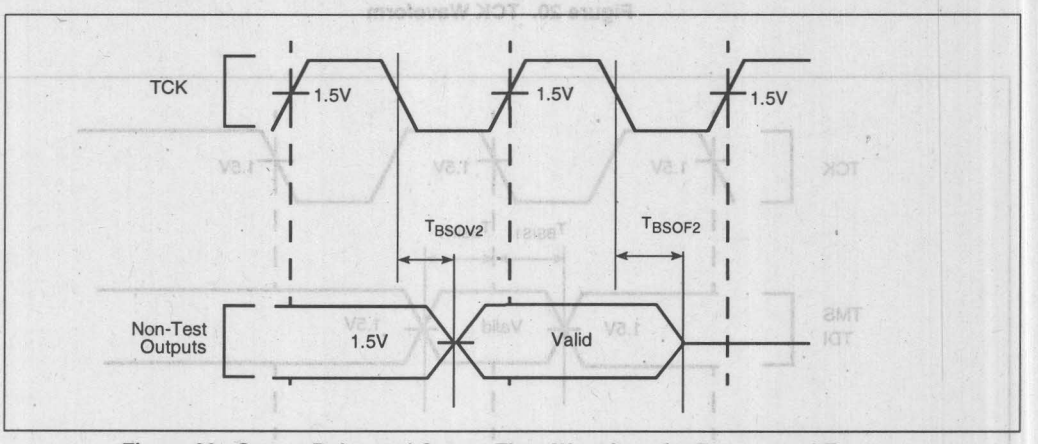


Figure 23. Output Delay and Output Float Waveform for T_{BSOV2} and T_{BSOF2}

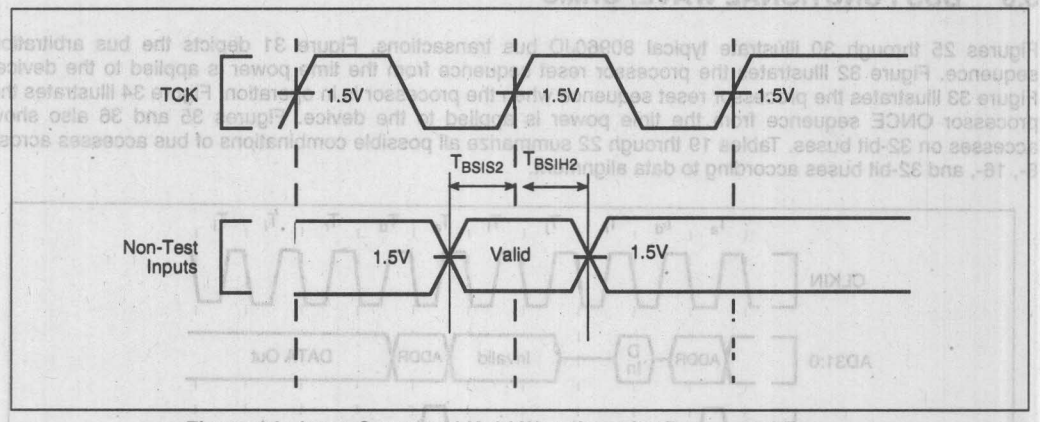


Figure 24. Input Setup and Hold Waveform for T_{BSIS2} and T_{BSIH2}

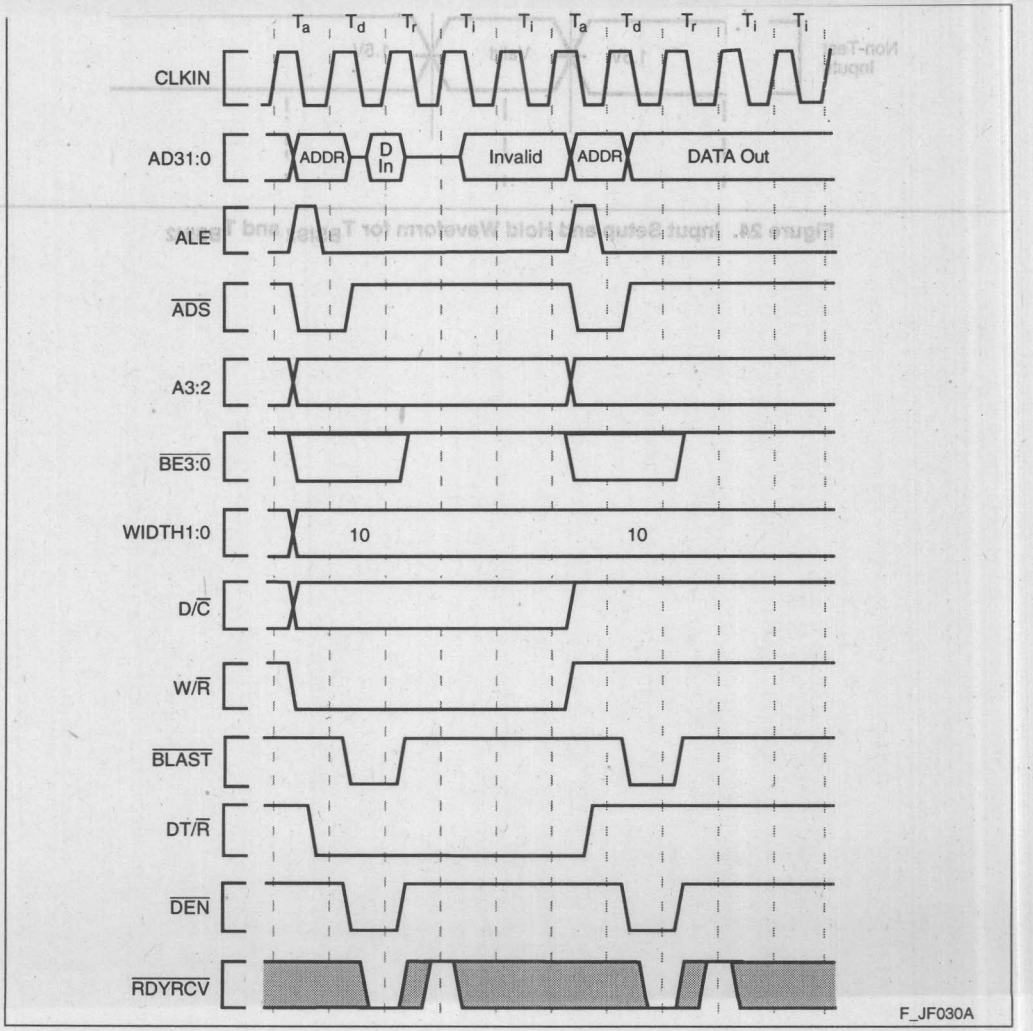


Figure 25. Non-Burst Read and Write Transactions Without Wait States, 32-Bit Bus

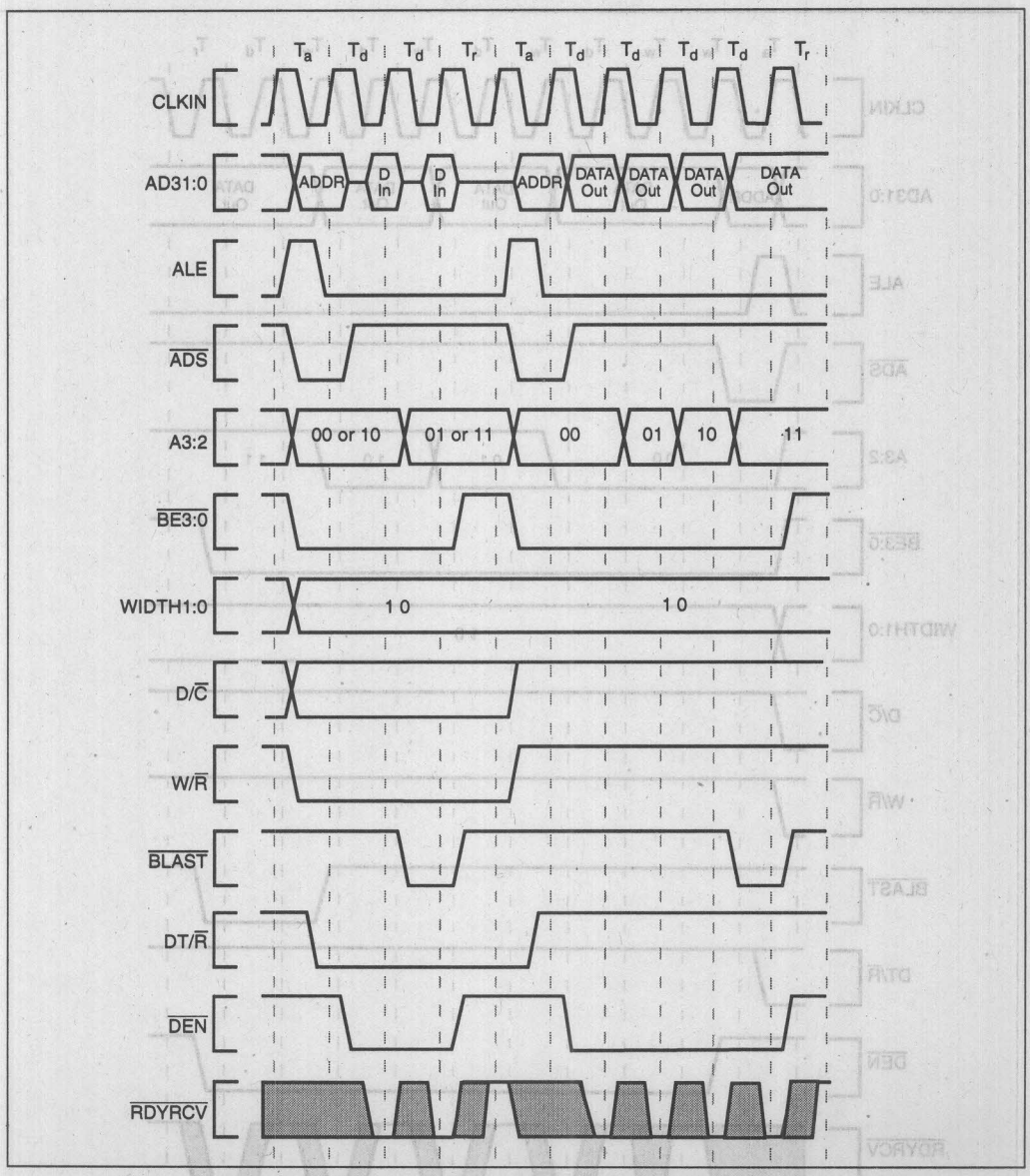


Figure 26. Burst Read and Write Transactions Without Wait States, 32-Bit Bus

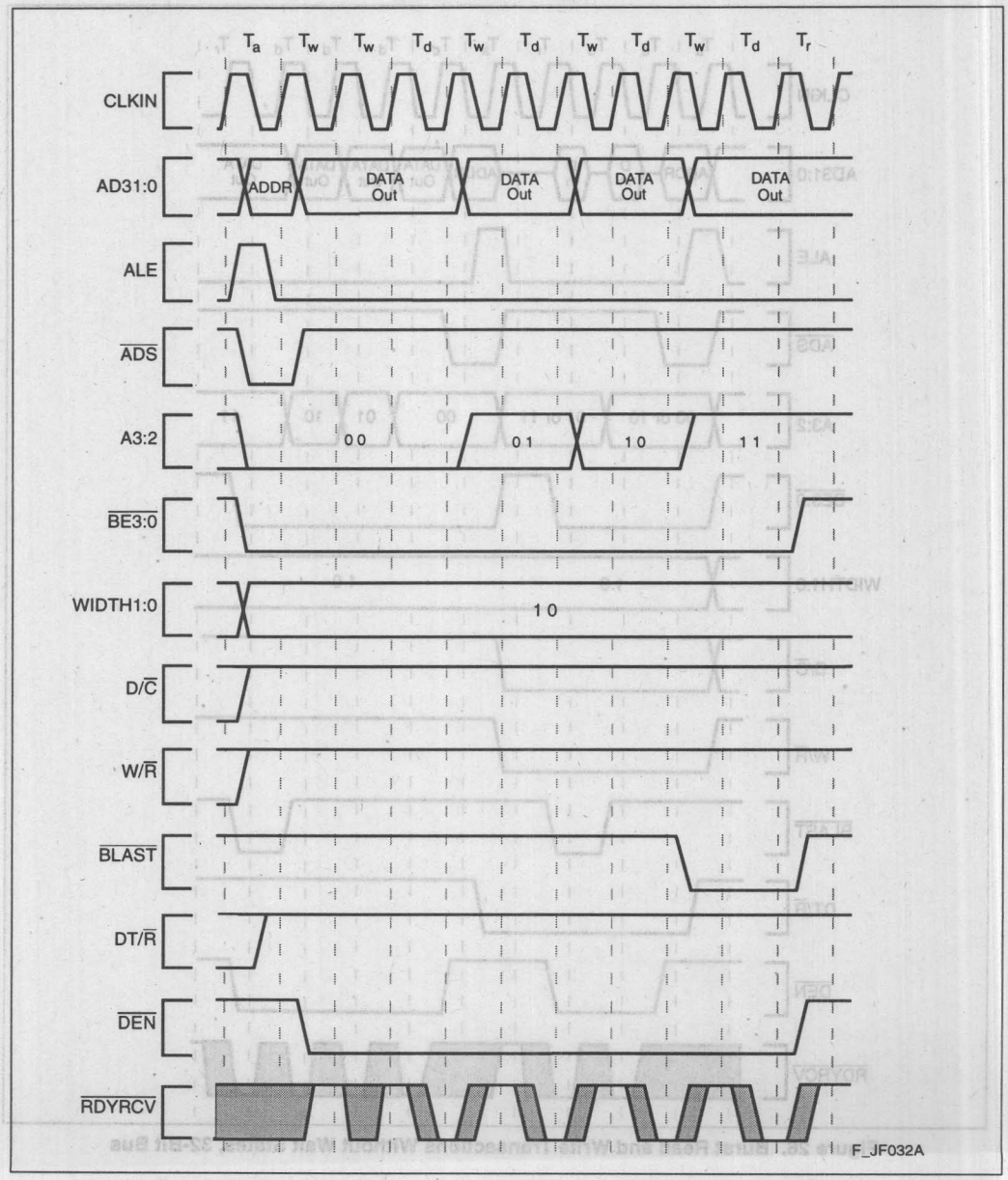


Figure 27. Burst Write Transactions With 2,1,1,1 Wait States, 32-Bit Bus

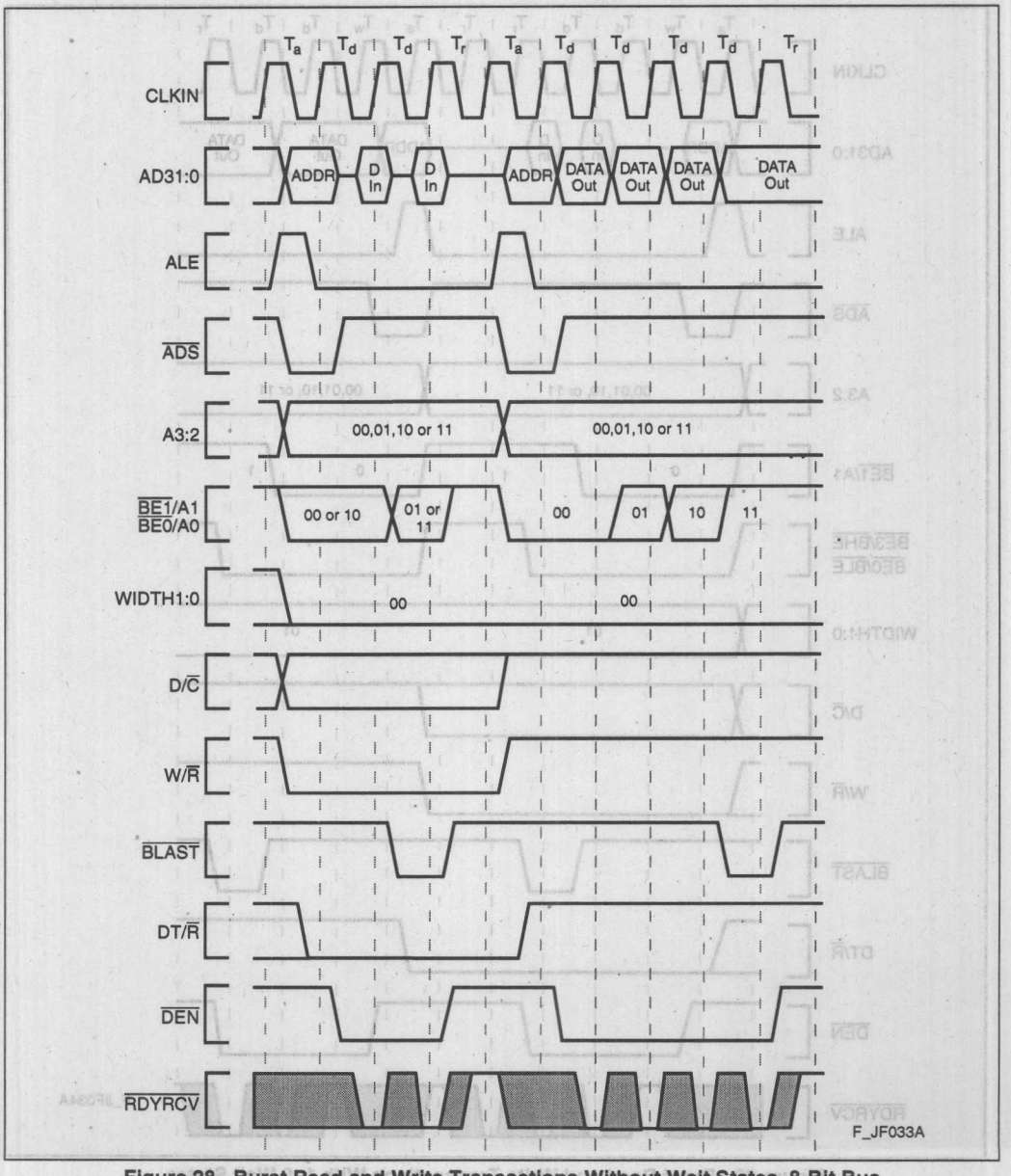


Figure 28. Burst Read and Write Transactions Without Wait States, 8-Bit Bus

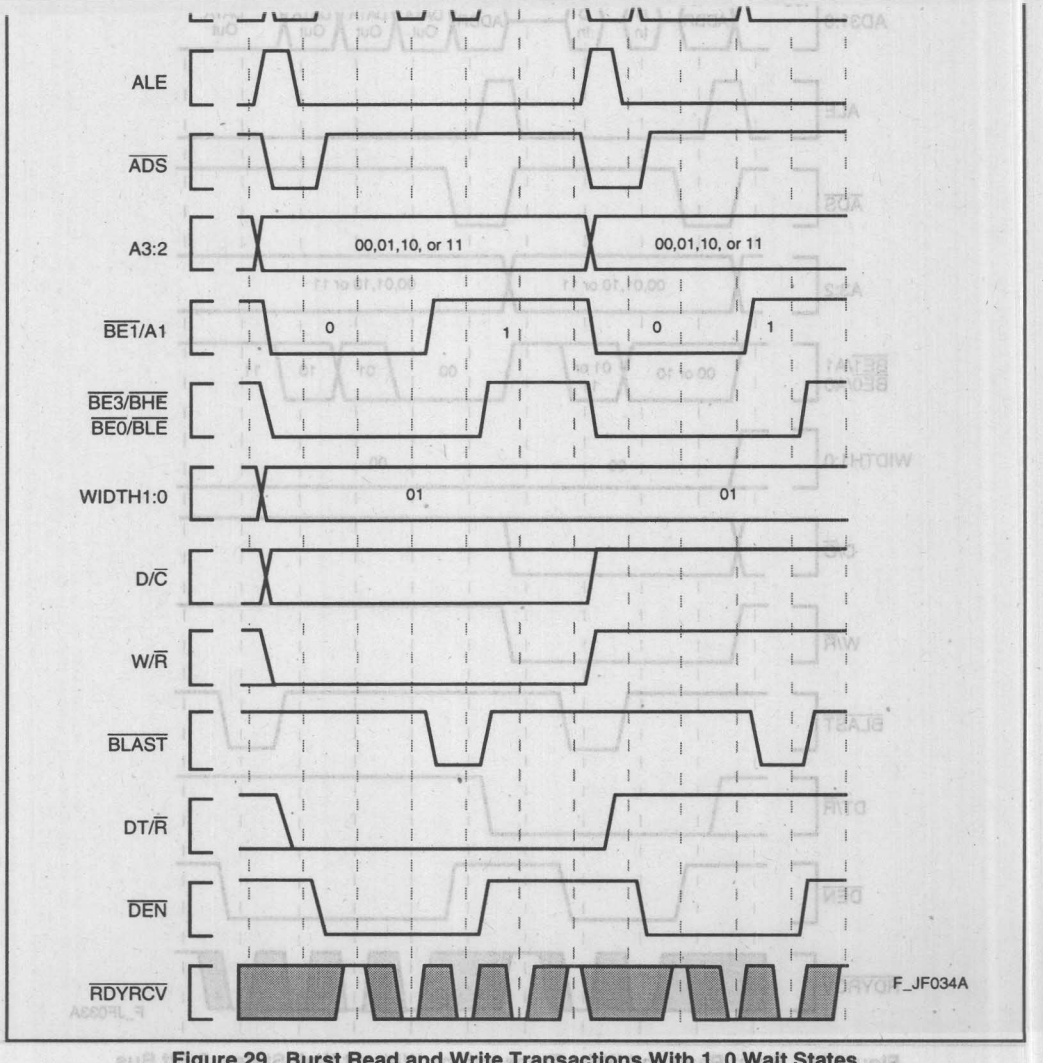


Figure 29. Burst Read and Write Transactions With 1, 0 Wait States and Extra Tr State on Read, 16-Bit Bus

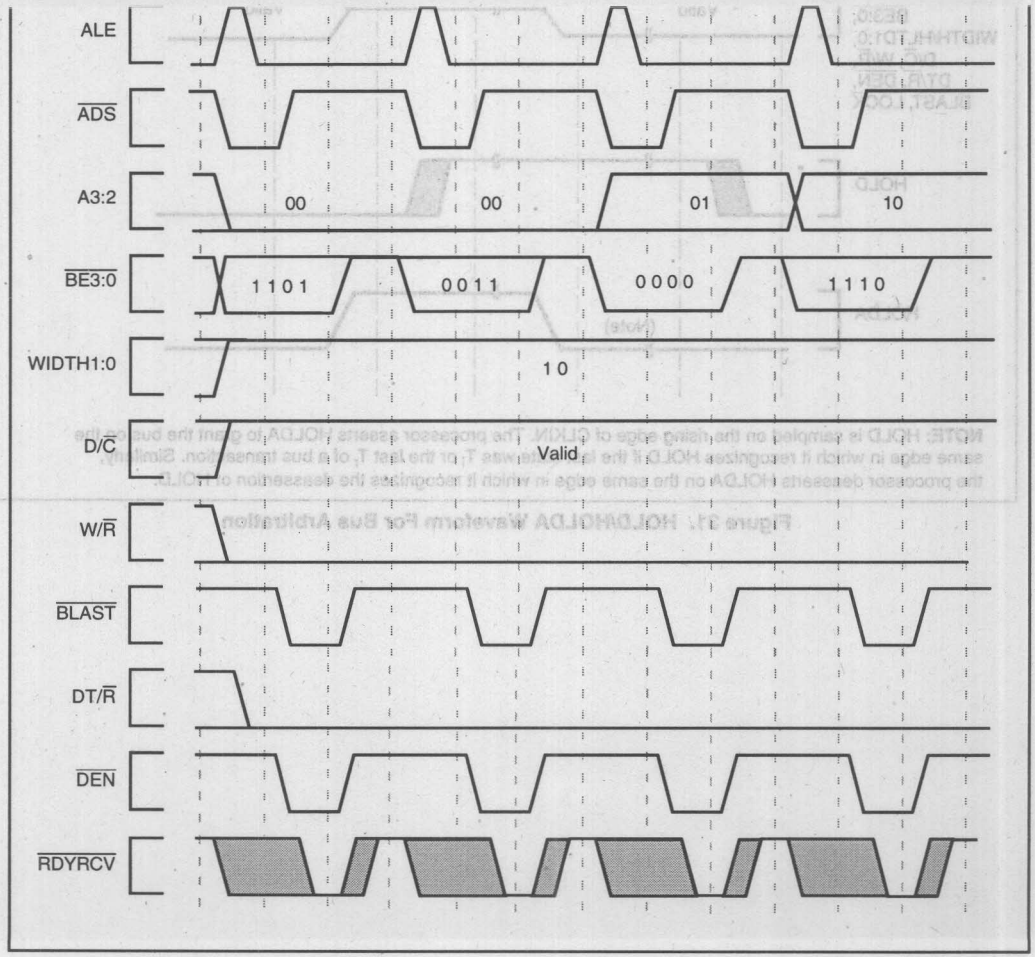
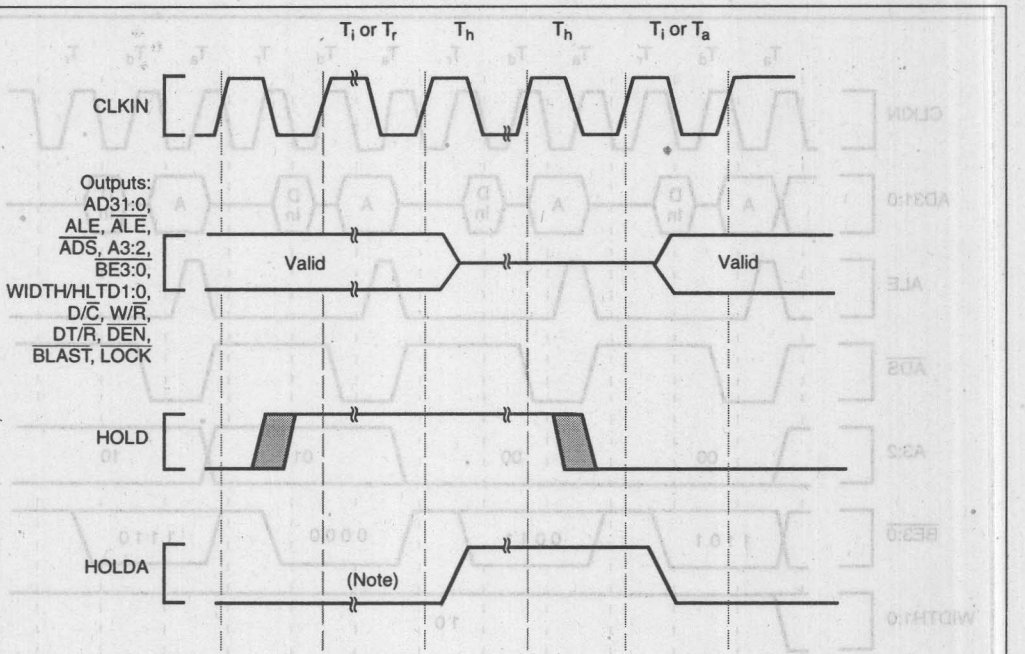


Figure 30. Bus Transactions Generated by Double Word Read Bus Request, Misaligned One Byte From Quad Word Boundary, 32-Bit Bus, Little Endian



NOTE: HOLD is sampled on the rising edge of CLKIN. The processor asserts HOLDA to grant the bus on the same edge in which it recognizes HOLD if the last state was T_i or the last T_r of a bus transaction. Similarly, the processor deasserts HOLDA on the same edge in which it recognizes the deassertion of HOLD.

Figure 31. HOLD/HOLDA Waveform For Bus Arbitration

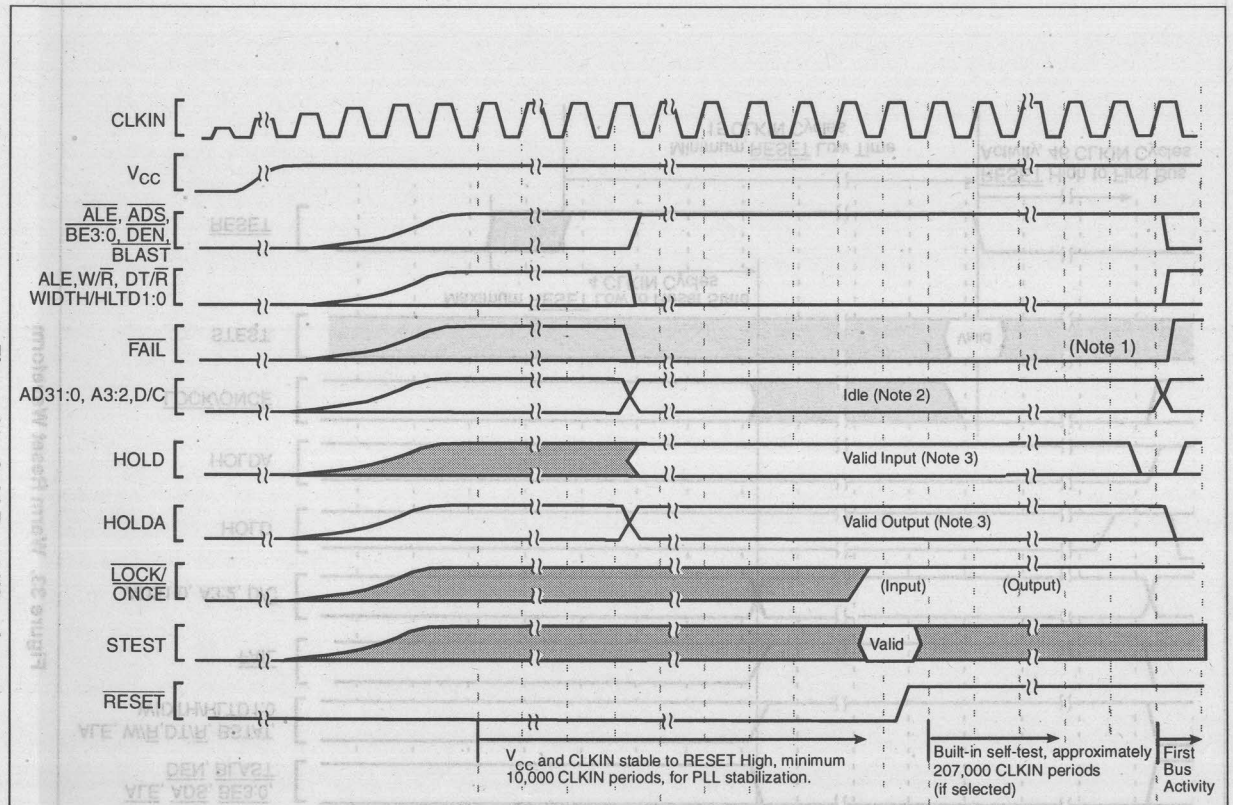


Figure 32. Cold Reset Waveform

Notes:

1. The processor asserts FAIL during built-in self-test. If self- test passes, the FAIL pin is deasserted.The processor also asserts FAIL during the bus confidence test. If the bus confidence test passes, FAIL is deasserted and the processor begins user program execution.
2. If the processor fails built-in self-test, it initiates one dummy load bus access. The load address indicates the point of self-test failure.
3. Since the bus is idle, hold requests are honored during reset and built-in self-test.

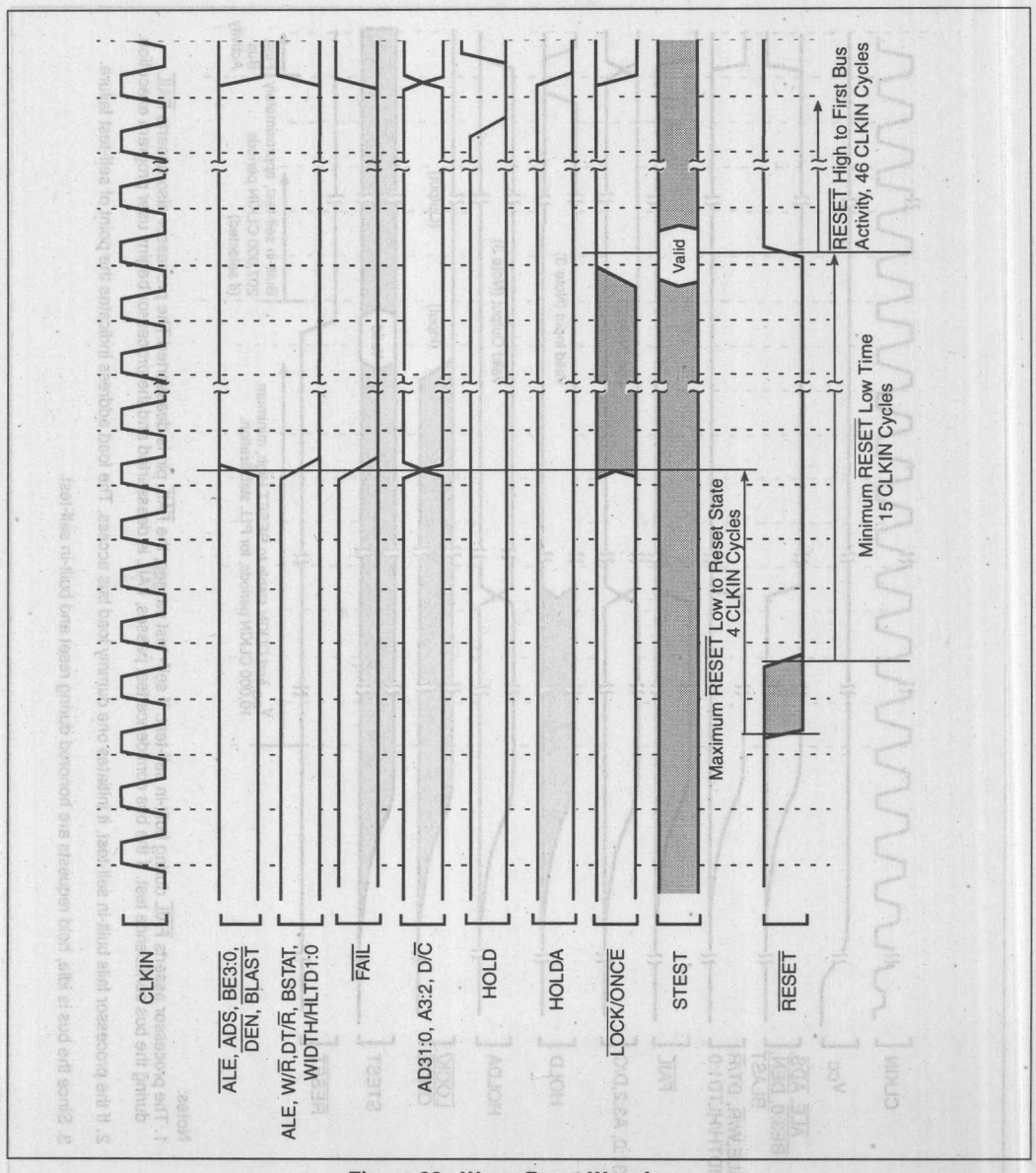


Figure 33. Warm Reset Waveform

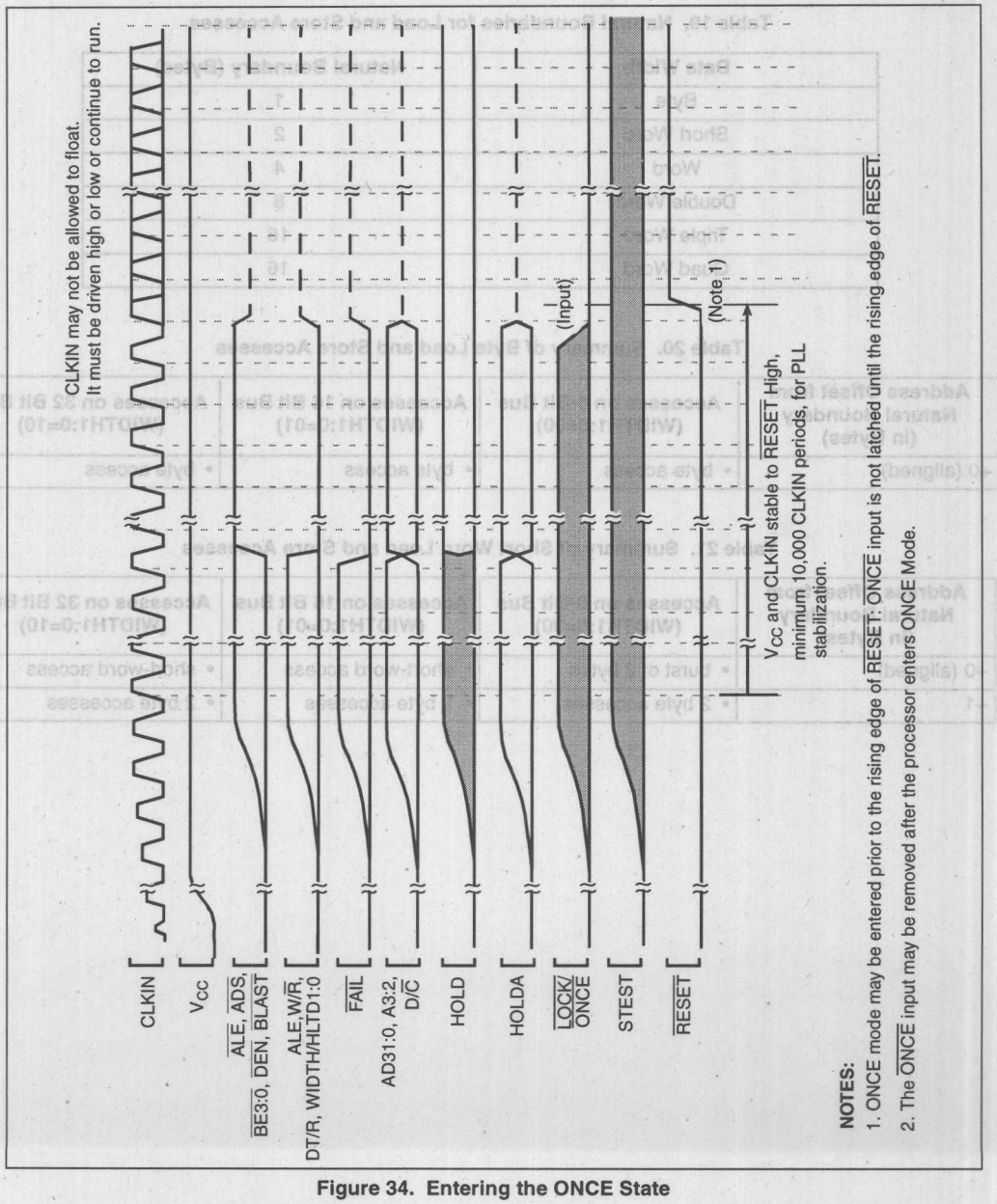


Figure 34. Entering the ONCE State

Triple Word	16
Quad Word	16

Table 20. Summary of Byte Load and Store Accesses

Address Offset from Natural Boundary (in Bytes)	Accesses on 8-Bit Bus (WIDTH1:0=00)	Accesses on 16 Bit Bus (WIDTH1:0=01)	Accesses on 32 Bit Bus (WIDTH1:0=10)
+0 (aligned)	• byte access	• byte access	• byte access

Table 21. Summary of Short Word Load and Store Accesses

Address Offset from Natural Boundary (in Bytes)	Accesses on 8-Bit Bus (WIDTH1:0=00)	Accesses on 16 Bit Bus (WIDTH1:0=01)	Accesses on 32 Bit Bus (WIDTH1:0=10)
+0 (aligned)	• burst of 2 bytes	• short-word access	• short-word access
+1	• 2 byte accesses	• 2 byte accesses	• 2 byte accesses

Table 22. Summary of *n*-Word Load and Store Accesses (*n* = 1, 2, 3, 4)

Address Offset from Natural Boundary In Bytes	Accesses on 8-Bit Bus (WIDTH1:0=00)	Accesses on 16 Bit Bus (WIDTH1:0=01)	Accesses on 32 Bit Bus (WIDTH1:0=10)
+0 (aligned) (*n* =1, 2, 3, 4)	• *n* burst(s) of 4 bytes	• case *n*=1: burst of 2 short words • case *n*=2: burst of 4 short words • case *n*=3: burst of 4 short words burst of 2 short words • case *n*=4: 2 bursts of 4 short words	• burst of *n* word(s)
+1 (*n* =1, 2, 3, 4) +5 (*n* = 2, 3, 4) +9 (*n* = 3, 4) +13 (*n* = 3, 4)	• byte access • burst of 2 bytes • *n*-1 burst(s) of 4 bytes • byte access	• byte access • short-word access • *n*-1 burst(s) of 2 short words • byte access	• byte access • short-word access • *n*-1 word access(es) • byte access
+2 (*n* =1, 2, 3, 4) +6 (*n* = 2, 3, 4) +10 (*n* = 3, 4) +14 (*n* = 3, 4)	• burst of 2 bytes • *n*-1 burst(s) of 4 bytes • burst of 2 bytes	• short-word access • *n*-1 burst(s) of 2 short words • short-word access	• short-word access • *n*-1 word access(es) • short-word access
+3 (*n* =1, 2, 3, 4) +7 (*n* = 2, 3, 4) +11 (*n* = 3, 4) +15 (*n* = 3, 4)	• byte access • *n*-1 burst(s) of 4 bytes • burst of 2 bytes • byte access	• byte access • *n*-1 burst(s) of 2 short words • short-word access • byte access	• byte access • *n*-1 word access(es) • short-word access • byte access
+4 (*n* = 2, 3, 4) +8 (*n* = 3, 4) +12 (*n* = 3, 4)	• *n* burst(s) of 4 bytes	• *n* burst(s) of 2 short words	• *n* word access(es)

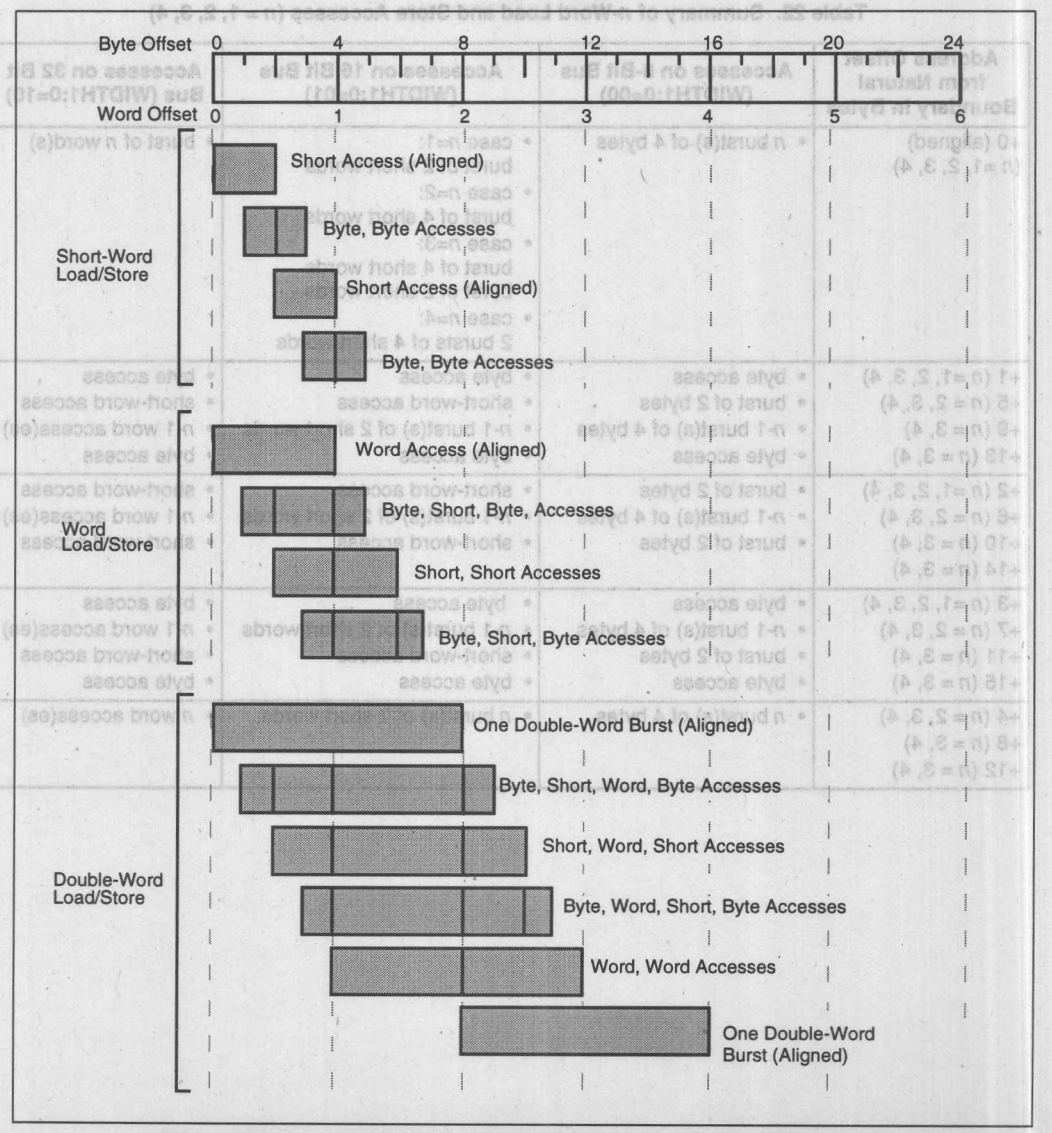


Figure 35. Summary of Aligned and Unaligned Accesses (32-Bit Bus)

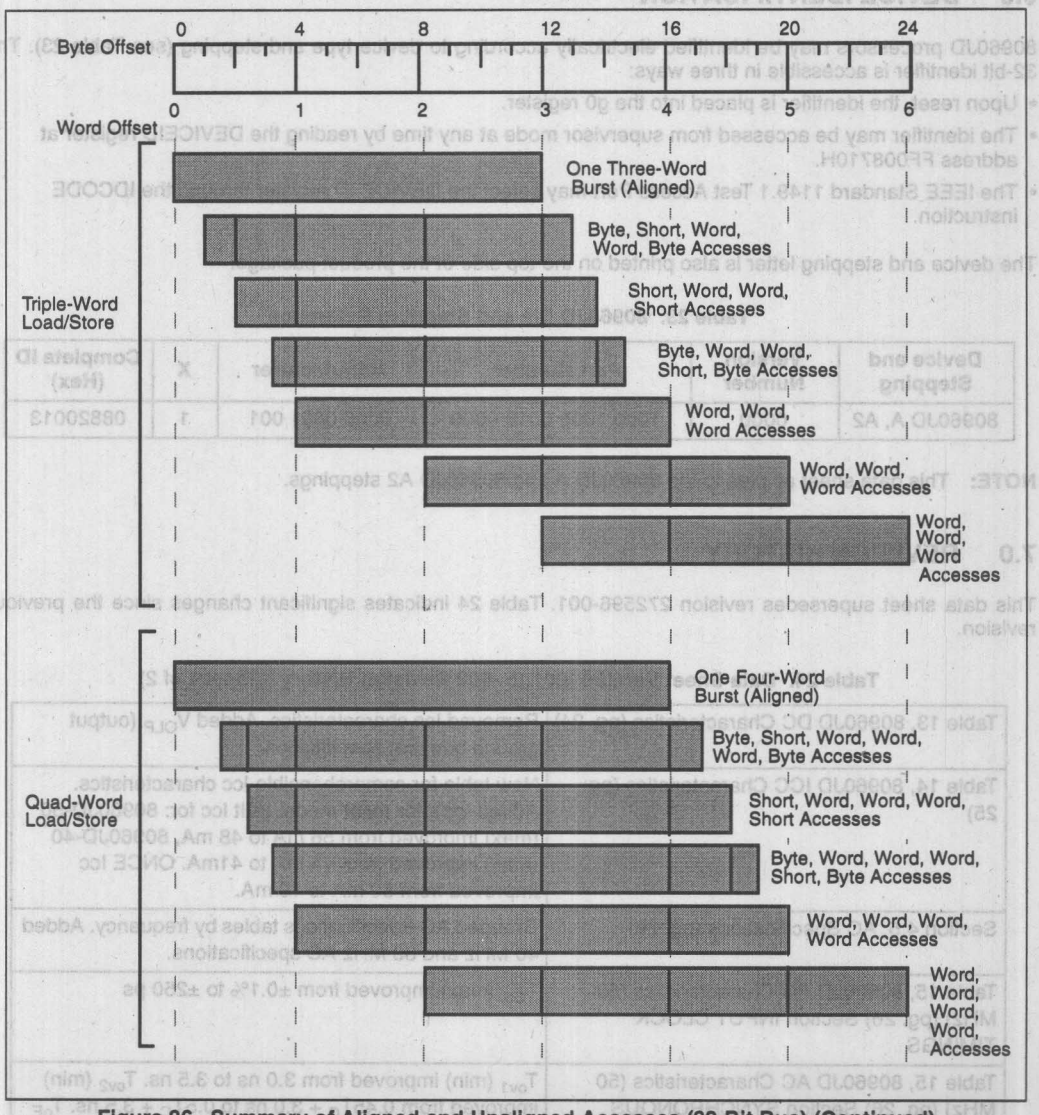


Figure 36. Summary of Aligned and Unaligned Accesses (32-Bit Bus) (Continued)

- The IEEE Standard 1149.1 Test Access Port may select the DEVICE ID register through the IDCODE instruction.

The device and stepping letter is also printed on the top side of the product package.

Table 23. 80960JD Die and Stepping Reference

Device and Stepping	Version Number	Part Number	Manufacturer	X	Complete ID (Hex)
80960JD A, A2	0000	1000 1000 0010 0000	0000 0001 001	1	08820013

NOTE: This data sheet applies to the 80960JD A and 80960JD A2 steppings.

7.0 REVISION HISTORY

This data sheet supersedes revision 272596-001. Table 24 indicates significant changes since the previous revision.

Table 24. Data Sheet Version -001 to -002 Revision History (Sheet 1 of 2)

Table 13, 80960JD DC Characteristics (pg. 24)	Removed Icc characteristics. Added V_{OLP} (output ground bounce) specification
Table 14, 80960JD ICC Characteristics (pg. 25)	New table for comprehensible Icc characteristics. Added Icc's for reset mode. Halt Icc for: 80960JD-50 (max) improved from 56 mA to 48 mA, 80960JD-40 (max) improved from 44 mA to 41mA. ONCE Icc Improved from 30 mA to 10 mA.
Section 4.5, AC Specifications (pg. 26)	Grouped AC Specifications tables by frequency. Added 40 MHz and 33 MHz AC specifications.
Table 15, 80960JD AC Characteristics (50 MHz) (pg. 26) Section INPUT CLOCK TIMINGS	T_{CS} (max) improved from ±0.1% to ±250 ps
Table 15, 80960JD AC Characteristics (50 MHz) (pg. 26) Section SYNCHRONOUS OUTPUT TIMINGS	T_{ov1} (min) improved from 3.0 ns to 3.5 ns. T_{ov2} (min) improved from $0.45T_C + 3.0$ ns to $0.5T_C + 3.5$ ns. T_{oF} (min) improved from 3.0 ns to 3.5 ns. T_{oF} (max) improved from 17 ns to 15 ns.

Table 24. Data Sheet Version -001 to -002 Revision History (Sheet 2 of 2)

Table 15, 80960JD AC Characteristics (50 MHz) (pg. 26) Section SYNCHRONOUS INPUT TIMINGS	T_{IS2} (min) improved from 10 ns to 9 ns
Table 15, 80960JD AC Characteristics (50 MHz) (pg. 26) Section RELATIVE OUTPUT TIMINGS	T_{LXL}, T_{LXA}, and T_{DXD} (min) improved from $.45T_C$ - 3 ns to $.5T_C$ - 7.5 ns.
Table 15, 80960JD AC Characteristics (50 MHz) (pg. 26) Section BOUNDARY SCAN TEST SIGNAL TIMINGS	T_{BSF} (max) improved from 8 MHz to $.5T_F$. T_{BSIS1} (min) improved from 8 ns to 4 ns. T_{BSIH1} (min) improved from 10 ns to 6 ns. T_{BSIS2} (min) improved from 8 ns to 4 ns. T_{BSIH2} (min) improved from 10 ns to 6 ns.

Table 24. Data Sheet Version -001 to -002 Revision History (Sheet 2 of 2)

Table 15, 80960JD AC Characteristics (50 MHz) (pg. 26) Section SYNCHRONOUS INPUT TIMINGS	T_{ds} (min) improved from 10 ns to 9 ns
Table 15, 80960JD AC Characteristics (50 MHz) (pg. 26) Section RELATIVE OUTPUT TIMINGS	T_{LXL}, T_{LXA}, and T_{DXD} (min) improved from $.45T_C$ - 3 ns to $.5T_C$ - 7.5 ns.
Table 15, 80960JD AC Characteristics (50 MHz) (pg. 26) Section BOUNDARY SCAN TEST SIGNAL TIMINGS	T_{BSF} (max) improved from 8 MHz to $.5T_F$, T_{BSS1} (min) improved from 8 ns to 4 ns. T_{BSH} (min) improved from 8 ns to 6 ns. T_{BSS2} (min) improved from 10 ns to 8 ns. T_{BSH2} (min) improved from 10 ns to 8 ns.

- 32-bit Parallel Architecture
  - Two Instructions/clock Execution
  - Load/Store Architecture
  - Sixteen 32-bit Global Registers
  - Sixteen 32-bit Local Registers
  - Manipulates 64-bit Bit Fields
  - 11 Addressing Modes
  - Full Parallel Fault Model
  - Supervisor Protection Model
- Fast Procedure Call/Return Model
  - Full Procedure Call in 4 Clocks
- On-Chip Register Cache
  - Caches Registers on Call/Ret
  - Minimum of 6 Frames Provided
  - Up to 15 Programmable Frames
- On-Chip Instruction Cache
  - 1 Kbyte Two-Way Set Associative
  - 128-bit Path to Instruction Sequencer
  - Cache-Lock Modes
  - Cache-Off Mode
- High Bandwidth On-Chip Data RAM
  - 1 Kbyte On-Chip Data RAM
  - Sustains 128 bits per Clock Access
- Four On-Chip DMA Channels
  - 59 Mbytes/s Fly-by Transfers
  - 32 Mbytes/s Two-Cycle Transfers
  - Data Chaining
  - Data Packing/Unpacking
  - Programmable Priority Method
- 32-Bit Demultiplexed Burst Bus
  - 128-bit Internal Data Paths to *and* from Registers
  - Burst Bus for DRAM Interfacing
  - Address Pipelining Option
  - Fully Programmable Wait States
  - Supports 8-, 16- or 32-bit Bus Widths
  - Supports Unaligned Accesses
  - Supervisor Protection Pin
- Selectable Big or Little Endian Byte Ordering
- High-Speed Interrupt Controller
  - Up to 248 External Interrupts
  - 32 Fully Programmable Priorities
  - Multi-mode 8-bit Interrupt Port
  - Four Internal DMA Interrupts
  - Separate, Non-maskable Interrupt Pin
  - Context Switch in 750 ns Typical

This complete document is available from Intel's World Wide Website and/or U.S. Literature Center:

World Wide Website: http://www.intel.com

U.S. Literature Center : 800-548-4725

in other geographies, please contact your local sales office

intel

PRELIMINARY

80960CF-40, -33, -25, -16
32-BIT HIGH-PERFORMANCE SUPERSCALAR EMBEDDED MICROPROCESSOR

• Socket and Object Code Compatible with 80960CA
• Two Instructions/Clock Sustained Execution
• Four 71 Mbytes/s DMA Channels with Data Chaining
• Demultiplexed 32-Bit Burst Bus with Pipelining

- **32-Bit Parallel Architecture**
  - **Two Instructions/clock Execution**
  - **Load/Store Architecture**
  - **Sixteen 32-Bit Global Registers**
  - **Sixteen 32-Bit Local Registers**
  - **Manipulates 64-Bit Bit Fields**
  - **11 Addressing Modes**
  - **Full Parallel Fault Model**
  - **Supervisor Protection Model**
- **Fast Procedure Call/Return Model**
  - **Full Procedure Call in 4 Clocks**
- **On-Chip Register Cache**
  - **Caches Registers on Call/Ret**
  - **Minimum of 6 Frames Provided**
  - **Up to 15 Programmable Frames**
- **On-Chip Instruction Cache**
  - **4 Kbyte Two-Way Set Associative**
  - **128-Bit Path to Instruction Sequencer**
  - **Cache-Lock Modes**
  - **Cache-Off Mode**
- **High Bandwidth On-Chip Data RAM**
  - **1 Kbyte On-Chip Data RAM**
  - **Sustains 128 bits per Clock Access**
- **Selectable Big or Little Endian Byte Ordering**
- **Four On-Chip DMA Channels**
  - **71 Mbytes/s Fly-by Transfers**
  - **40 Mbytes/s Two-Cycle Transfers**
  - **Data Chaining**
  - **Data Packing/Unpacking**
  - **Programmable Priority Method**
- **32-Bit Demultiplexed Burst Bus**
  - **128-Bit Internal Data Paths to *and* from Registers**
  - **Burst Bus for DRAM Interfacing**
  - **Address Pipelining Option**
  - **Fully Programmable Wait States**
  - **Supports 8-, 16- or 32-Bit Bus Widths**
  - **Supports Unaligned Accesses**
  - **Supervisor Protection Pin**
- **High-Speed Interrupt Controller**
  - **Up to 248 External Interrupts**
  - **32 Fully Programmable Priorities**
  - **Multi-mode 8-Bit Interrupt Port**
  - **Four Internal DMA Interrupts**
  - **Separate, Non-maskable Interrupt Pin**
  - **Context Switch in 625 ns Typical**
- **On-Chip Data Cache**
  - **1 Kbyte Direct-Mapped, Write Through**
  - **128 bits per Clock Access on Cache Hit**

June 1996

Order Number: 272886-001

intel

PRELIMINARY

80960CF-40, -33, -25, -16
32-BIT HIGH-PERFORMANCE SUPERSCALAR EMBEDDED MICROPROCESSOR

- Socket and Object Code Compatible with 80960CA
- Two Instructions/Clock Sustained Execution
- Four 71 Mbytes/s DMA Channels with Data Chaining
- Demultiplexed 32-Bit Burst Bus with Pipelining

■ 32-Bit Parallel Architecture
— Two Instructions/clock Execution
— Load/Store Architecture
— Sixteen 32-Bit Global Registers
— Sixteen 32-Bit Local Registers
— Manipulates 64-Bit Bit Fields
— 11 Addressing Modes
— Full Parallel Fault Model
— Supervisor Protection Model

■ Fast Procedure Call/Return Model
— Full Procedure Call in 4 Clocks

■ On-Chip Register Cache
— Caches Registers on Call/Ret
— Minimum of 6 Frames Provided
— Up to 15 Programmable Frames

■ On-Chip Instruction Cache
— 4 Kbyte Two-Way Set Associative
— 128-Bit Path to Instruction Sequencer
— Cache-Lock Modes
— Cache-Off Mode

■ High Bandwidth On-Chip Data RAM
— 1-Kbyte On-Chip Data RAM
— Sustains 128 bits per Clock Access

■ Selectable Big or Little Endian Byte Ordering

■ Four On-Chip DMA Channels
— 71 Mbytes/s Fly-by Transfers
— 40 Mbytes/s Two-Cycle Transfers
— Data Chaining
— Data Packing/Unpacking
— Programmable Priority Method

■ 32-Bit Demultiplexed Burst Bus
— 128-Bit Internal Data Paths to and from Register
— Burst Bus for DRAM Interfacing
— Address Pipelining Option
— Fully Programmable Wait States
— Supports 8-, 16- or 32-Bit Bus Widths
— Supports Unaligned Accesses
— Supervisor Protection Pin

■ High-Speed Interrupt Controller
— Up to 248 External Interrupts
— 32 Fully Programmable Priorities
— Multi-mode 8-Bit Interrupt Port
— Four Internal DMA Interrupts
— Separate, Non-maskable Interrupt Pin
— Context Switch in 625 ns Typical

■ On-Chip Data Cache
— 1 Kbyte Direct-Mapped, Write Through
— 128 bits per Clock Access on Cache Hit

June 1995 Order Number: 272865-001

worldwide	916-356-3105
Application Bulletin Board Service:	
up to 14.4-Kbaud line, worldwide	916-356-3600
dedicated 2400-baud line, worldwide	916-356-7209
Europe	+44(0)793-496340

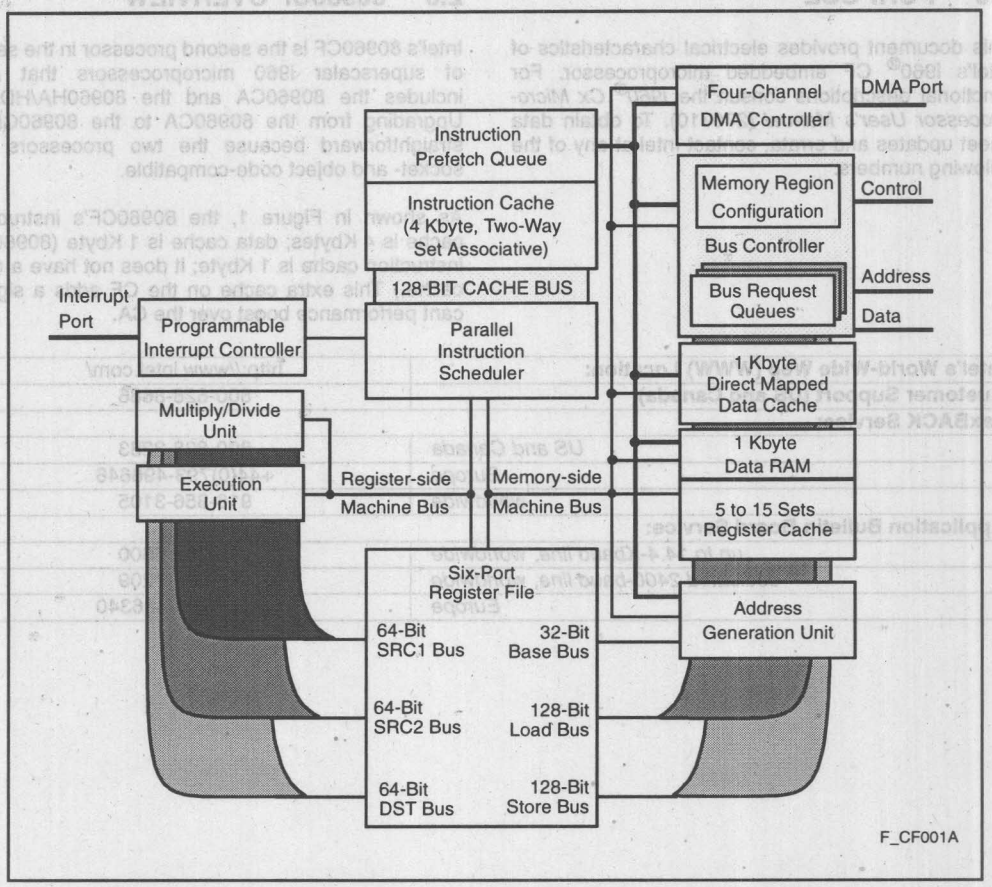


Figure 1. 80960CF Block Diagram

The 80960CF, object code compatible with the 32-bit 80960 core Architecture, employs Special Function Register extensions to control on-chip peripherals and instruction set extensions to shift 64-bit operands and configure on-chip hardware. Multiple 128-bit internal buses, on-chip instruction caching and a sophisticated instruction scheduler allow the processor to sustain execution of two instructions per clock with peak execution of three instructions per clock.

A 32-bit demultiplexed and pipelined burst bus provides a 132 Mbyte/s bandwidth to a system's high-speed external memory subsystem. Also, the 80960CF's on-chip caching of instructions, procedure context and critical program data substantially decouples system performance from the wait states associated with accesses to the system's slower, cost sensitive, main memory subsystem.

The 80960CF bus controller integrates full wait state and bus width control for highest system performance with minimal system design complexity. Unaligned access and Big Endian byte order support reduces the cost of porting existing applications to the 80960CF.

The processor also integrates four complete data-chaining DMA channels and a high-speed interrupt controller on-chip. DMA channels perform single-cycle or two-cycle transfers, data packing and unpacking and data chaining. Block transfers — in addition to source or destination synchronized transfers — are supported.

The interrupt controller provides full programmability of 248 interrupt sources into 32 priority levels with a typical interrupt task switch (latency) time of 625 ns.

2.1 The 80960C-Series Core

The C-Series core is a very high performance microarchitectural implementation of the 80960 Core Architecture. This core can sustain execution of two instructions per clock (80 MIPS at 40 MHz). To achieve this level of performance, Intel has incorporated state-of-the-art silicon technology and innovative microarchitectural constructs into the C-Series core implementation. Factors that contribute to the core's performance include:

- Parallel instruction decoding allows issuance of up to three instructions per clock
- Single-clock execution of most instructions
- Parallel instruction decode allows sustained, simultaneous execution of two single-clock instructions every clock cycle
- Efficient instruction pipeline minimizes pipeline break losses
- Register and resource scoreboarding allow simultaneous multi-clock instruction execution
- Branch look-ahead and prediction allows many branches to execute with no pipeline break
- Local Register Cache integrated on-chip caches Call/Return context
- Two-way set associative, 4 Kbyte integrated instruction cache
- 1 Kbyte integrated Data RAM sustains a four-word (128-bit) access every clock cycle
- Direct mapped, 1 Kbyte data cache, write through, write allocate

2.2 Pipelined, Burst Bus

A 32-bit high performance bus controller interfaces the 80960CF to external memory and peripherals. The Bus Control Unit features a maximum transfer rate of 160 Mbytes per second (at 40 MHz). Internally programmable wait states and 16 separately configurable memory regions allow the processor to interface with a variety of memory subsystems with a minimum of system complexity and a maximum of performance. The Bus Control Unit's main features include:

- Demultiplexed, burst bus to exploit most efficient DRAM access modes
- Address pipelining to reduce memory cost while maintaining performance
- 32-, 16- and 8-bit modes for I/O interfacing ease
- Full internal wait state generation to reduce system cost
- Little and Big Endian support to ease application development
- Unaligned access support for code portability
- Three-deep request queue to decouple the bus from the core

2.3 Instruction Set Summary

Table 1 summarizes the 80960CF instruction set by logical groupings. See the *i960® Cx Microprocessor User's Manual* (270710) for a complete description of the instruction set.

2.4 Flexible DMA Controller

A four-channel DMA controller provides high speed DMA control for data transfers involving peripherals and memory. The DMA provides advanced features such as data chaining, byte assembly and disassembly and a high performance fly-by mode capable of transfer speeds of up to 71 Mbytes per second at 40 MHz. The DMA controller features a performance and flexibility which is only possible by integrating the DMA controller and the 80960CF core.

Data Movement	Arithmetic	Logical	Bit / Bit Field / Byte
Load	Add	And	Set Bit
Store	Subtract	Not And	Clear Bit
Move	Multiply	And Not	Not Bit
Load Address	Divide	Or	Alter Bit
	Remainder	Exclusive Or	Scan For Bit
	Modulo	Not Or	Span Over Bit
	Shift	Or Not	Extract
	*Extended Shift	Nor	Modify
	Extended Multiply	Exclusive Nor	Scan Byte for Equal
	Extended Divide	Not	
	Add with Carry	Nand	
	Subtract with Carry		
	Rotate		
Comparison	**Branch**	**Call/Return**	**Fault**
Compare	Unconditional Branch	Call	Conditional Fault
Conditional Compare	Conditional Branch	Call Extended	Synchronize Faults
Compare and Increment	Compare and Branch	Call System	
Compare and Decrement		Return	
Test Condition Code		Branch and Link	
Check Bit			
Debug	**Processor Mgmt**	**Atomic**	
Modify Trace Controls	Flush Local Registers	Atomic Add	
Mark	Modify Arithmetic Controls	Atomic Modify	
Force Mark	Modify Process Controls		
	*System Control		
	*DMA Control		

NOTES: Instructions marked by (*) are 80960Cx extensions to the 80960 instruction set.

3.0 PACKAGE INFORMATION

3.1 Package Introduction

This section describes the pins, pinouts and thermal characteristics for the 80960CF in the 168-pin Ceramic Pin Grid Array (PGA) package; the 80960CF-33, -25, -16 devices are also available in the 196-pin Plastic Quad Flat Package (PQFP). For complete package specifications and information, see the *Packaging* Handbook (# 240800).

3.2 Pin Descriptions

This section defines the 80960CF pins. Table 2 presents the legend for interpreting the pin descriptions in the following tables. Pins associated with the 32-bit demultiplexed processor bus are described in Table 2. Pins associated with the 80960CF DMA Controller and Interrupt Unit are described in Table 3. Pins associated with basic processor configuration and control are described in Table 2.

All pins float while the processor is in the ONCE mode.

Symbol	Description
I	Input only pin
O	Output only pin
I/O	Pin can be either an input or output
–	Pins "must be" connected as described
S(...)	Synchronous. Inputs must meet setup and hold times relative to PCLK2:1 for proper operation. Outputs are synchronous to PCLK2:1. S(E) Edge sensitive input S(L) Level sensitive input
A(...)	Asynchronous. Inputs may be asynchronous to PCLK2:1. A(E) Edge sensitive input A(L) Level sensitive input
H(...)	While the bus is in the Hold Acknowledge or Bus Backoff state, the pin: H(1) is driven to V_{CC} H(0) is driven to V_{SS} H(Z) floats H(Q) continues to be a valid input
R(...)	While the processor's $\overline{RESET}$ pin is low, the pin: R(1) is driven to V_{CC} R(0) is driven to V_{SS} R(Z) floats R(Q) continues to be a valid output

Table 2. 80960CF Pin Description — External Bus Signals (Sheet 1 of 3)

Name	Type	Description
A31:2	O S H(Z) R(Z)	**ADDRESS BUS** carries the physical address' upper 30 bits. A31 is the most significant bit; A2 is least significant. During a bus access, A31:2 identify all external addresses to word (4-byte) boundaries. Byte enable signals indicate the selected byte in each word. During burst accesses, A3:2 increment to indicate successive data cycles.
D31:0	I/O S(L) H(Z) R(Z)	**DATA BUS** carries 32-, 16- or 8-bit data quantities depending on bus width configuration. The least significant bit is carried on D0 and the most significant on D31. When the bus is configured for 8-bit data, the lower 8 data lines, D7:0 are used. For 16-bit data bus widths, D15:0 are used. For 32-bit bus widths the full data bus is used.
$\overline{BE3:0}$	O S H(Z) R(1)	**BYTE ENABLES** select which of the four bytes addressed by A31:2 are active during an access to a memory region configured for a 32-bit data-bus width. $\overline{BE3}$ applies to D31:24; $\overline{BE2}$ applies to D23:16; $\overline{BE1}$ applies to D15:8 $\overline{BE0}$ applies to D7:0. 32-bit bus: $\overline{BE3}$ Byte Enable 3 enable D31:24 $\overline{BE2}$ Byte Enable 2 enable D23:16 $\overline{BE1}$ Byte Enable 1 enable D15:8 $\overline{BE0}$ Byte Enable 0 enable D7:0 For accesses to a memory region configured for a 16-bit data-bus width, the processor uses the $\overline{BE3}$, $\overline{BE1}$ and $\overline{BE0}$ pins as $\overline{BHE}$, A1 and $\overline{BLE}$ respectively. 16-bit bus: $\overline{BE3}$ Byte High Enable ($\overline{BHE}$) enable D15:8 $\overline{BE2}$ Not used (driven high or low) $\overline{BE1}$ Address Bit 1 (A1) $\overline{BE0}$ Byte Low Enable ($\overline{BLE}$) enable D7:0 For accesses to a memory region configured for an 8-bit data-bus width, the processor uses the $\overline{BE1}$ and $\overline{BE0}$ pins as A1 and A0 respectively. 8-bit bus: $\overline{BE3}$ Not used (driven high or low) $\overline{BE2}$ Not used (driven high or low) $\overline{BE1}$ Address Bit 1 (A1) $\overline{BE0}$ Address Bit 0 (A0)
W/R	O S H(Z) R(0)	**WRITE/READ** is asserted for read requests and deasserted for write requests. The W/$\overline{R}$ signal changes in the same clock cycle as $\overline{ADS}$. It remains valid for the entire access in non-pipelined regions. In pipelined regions, W/$\overline{R}$ is not guaranteed to be valid in the last cycle of a read access.
$\overline{ADS}$	O S H(Z) R(1)	**ADDRESS STROBE** indicates a valid address and the start of a new bus access. $\overline{ADS}$ is asserted for the first clock of a bus access.
$\overline{READY}$	I S(L) H(Z) R(Z)	**READY** is an input which signals the termination of a data transfer. $\overline{READY}$ is used to indicate that read data on the bus is valid or that a write-data transfer has completed. The $\overline{READY}$ signal works in conjunction with the internally programmed wait-state generator. If $\overline{READY}$ is enabled in a region, the pin is sampled after the programmed number of wait-states has expired. If the $\overline{READY}$ pin is deasserted, wait states continue to be inserted until $\overline{READY}$ becomes asserted. This is true for the N_{RAD}, N_{RDD}, N_{WAD} and N_{WDD} wait states. The N_{XDA} wait states cannot be extended.

Table 2. 80960CF Pin Description — External Bus Signals (Sheet 2 of 3)

Name	Type	Description
$\overline{\text{BTERM}}$	I S(L) H(Z) R(Z)	**BURST TERMINATE** is an input which breaks up a burst access and causes another address cycle to occur. The $\overline{\text{BTERM}}$ signal works in conjunction with the internally programmed wait-state generator. If $\overline{\text{READY}}$ and $\overline{\text{BTERM}}$ are enabled in a region, the $\overline{\text{BTERM}}$ pin is sampled after the programmed number of wait states has expired. When $\overline{\text{BTERM}}$ is asserted, a new $\overline{\text{ADS}}$ signal is generated and the access is completed. The $\overline{\text{READY}}$ input is ignored when $\overline{\text{BTERM}}$ is asserted. $\overline{\text{BTERM}}$ must be externally synchronized to satisfy $\overline{\text{BTERM}}$ setup and hold times.
$\overline{\text{WAIT}}$	O S H(Z) R(1)	**WAIT** indicates internal wait state generator status. $\overline{\text{WAIT}}$ is asserted when wait states are being caused by the internal wait state generator and not by the $\overline{\text{READY}}$ or $\overline{\text{BTERM}}$ inputs. $\overline{\text{WAIT}}$ can be used to derive a write-data strobe. $\overline{\text{WAIT}}$ can also be thought of as a $\overline{\text{READY}}$ output that the processor provides when it is inserting wait states.
$\overline{\text{BLAST}}$	O S H(Z) R(0)	**BURST LAST** indicates the last transfer in a bus access. $\overline{\text{BLAST}}$ is asserted in the last data transfer of burst and non-burst accesses after the wait state counter reaches zero. $\overline{\text{BLAST}}$ remains asserted until the clock following the last cycle of the last data transfer of a bus access. If the $\overline{\text{READY}}$ or $\overline{\text{BTERM}}$ input is used to extend wait states, the $\overline{\text{BLAST}}$ signal remains asserted until $\overline{\text{READY}}$ or $\overline{\text{BTERM}}$ terminates the access.
DT/$\overline{\text{R}}$	O S H(Z) R(0)	**DATA TRANSMIT/RECEIVE** indicates direction for data transceivers. DT/$\overline{\text{R}}$ is used in conjunction with $\overline{\text{DEN}}$ to provide control for data transceivers attached to the external bus. When DT/$\overline{\text{R}}$ is asserted, the signal indicates that the processor receives data. Conversely, when deasserted, the processor sends data. DT/$\overline{\text{R}}$ changes only while $\overline{\text{DEN}}$ is high.
$\overline{\text{DEN}}$	O S H(Z) R(1)	**DATA ENABLE** indicates data cycles in a bus request. $\overline{\text{DEN}}$ is asserted at the start of the bus request first data cycle and is deasserted at the end of the last data cycle. $\overline{\text{DEN}}$ is used in conjunction with DT/$\overline{\text{R}}$ to provide control for data transceivers attached to the external bus. $\overline{\text{DEN}}$ remains asserted for sequential reads from pipelined memory regions. $\overline{\text{DEN}}$ is deasserted when DT/$\overline{\text{R}}$ changes.
$\overline{\text{LOCK}}$	O S H(Z) R(1)	**BUS LOCK** indicates that an atomic read-modify-write operation is in progress. $\overline{\text{LOCK}}$ may be used to prevent external agents from accessing memory which is currently involved in an atomic operation. $\overline{\text{LOCK}}$ is asserted in the first clock of an atomic operation and deasserted in the clock cycle following the last bus access for the atomic operation. To allow the most flexibility for memory system enforcement of locked accesses, the processor acknowledges a bus hold request when $\overline{\text{LOCK}}$ is asserted. The processor performs DMA transfers while $\overline{\text{LOCK}}$ is active.
HOLD	I S(L) H(Z) R(Z)	**HOLD REQUEST** signals that an external agent requests access to the external bus. The processor asserts HOLDA after completing the current bus request. HOLD, HOLDA and BREQ are used together to arbitrate access to the processor's external bus by external bus agents.
$\overline{\text{BOFF}}$	I S(L) H(Z) R(Z)	**BUS BACKOFF**, when asserted, suspends the current access and causes the bus pins to float. When $\overline{\text{BOFF}}$ is deasserted, the $\overline{\text{ADS}}$ signal is asserted on the next clock cycle and the access is resumed.

Table 2. 80960CF Pin Description — External Bus Signals (Sheet 3 of 3)

Name	Type	Description
HOLDA	O S H(1) R(Q)	**HOLD ACKNOWLEDGE** indicates to a bus requestor that the processor has relinquished control of the external bus. When HOLDA is asserted, the external address bus, data bus and bus control signals are floated. HOLD, $\overline{\text{BOFF}}$, HOLDA and BREQ are used together to arbitrate access to the processor's external bus by external bus agents. Since the processor grants HOLD requests and enters the Hold Acknowledge state even while $\overline{\text{RESET}}$ is asserted, the state of the HOLDA pin is independent of the $\overline{\text{RESET}}$ pin.
BREQ	O S H(Q) R(0)	**BUS REQUEST** is asserted when the bus controller has a request pending. BREQ can be used by external bus arbitration logic in conjunction with HOLD and HOLDA to determine when to return mastership of the external bus to the processor.
D/$\overline{\text{C}}$	O S H(Z) R(Z)	**DATA OR CODE** is asserted for a data request and deasserted for instruction requests. D/$\overline{\text{C}}$ has the same timing as W/$\overline{\text{R}}$.
$\overline{\text{DMA}}$	O S H(Z) R(Z)	**DMA ACCESS** indicates whether the bus request was initiated by the DMA controller. $\overline{\text{DMA}}$ is asserted for any DMA request. $\overline{\text{DMA}}$ is deasserted for all other requests.
$\overline{\text{SUP}}$	O S H(Z) R(Z)	**SUPERVISOR ACCESS** indicates whether the bus request is issued while in supervisor mode. $\overline{\text{SUP}}$ is asserted when the request has supervisor privileges and is deasserted otherwise. $\overline{\text{SUP}}$ can be used to isolate supervisor code and data structures from non-supervisor requests.

Table 3. 80960CF Pin Description — Processor Control Signals (Sheet 1 of 2)

Name	Type	Description
$\overline{\text{RESET}}$	I A(L) H(Z) R(Z)	**RESET** causes the chip to reset. When $\overline{\text{RESET}}$ is asserted, all external signals return to the reset state. When $\overline{\text{RESET}}$ is deasserted, initialization begins. When the 2-x clock mode is selected, $\overline{\text{RESET}}$ must remain asserted for 32 CLKIN cycles before being deasserted to guarantee correct processor initialization. When the 1-x clock mode is selected, $\overline{\text{RESET}}$ must remain asserted for 10,000 CLKIN cycles before being deasserted to guarantee correct processor initialization. The CLKMODE pin selects 1-x or 2-x input clock division of the CLKIN pin. The Hold Acknowledge bus state functions while the chip is reset. If the bus is in the Hold Acknowledge state when $\overline{\text{RESET}}$ is asserted, the processor internally resets, but maintains the Hold Acknowledge state on external pins until the Hold request is removed. If a Hold request is made while the processor is in the reset state, the processor bus grants HOLDA and enters the Hold Acknowledge state.
$\overline{\text{FAIL}}$	O S H(Q) R(0)	**FAIL** indicates failure of the self-test performed at initialization. When $\overline{\text{RESET}}$ is deasserted and initialization begins, the $\overline{\text{FAIL}}$ pin is asserted. An internal self-test is performed as part of the initialization process. If this self-test passes, the $\overline{\text{FAIL}}$ pin is deasserted; otherwise it remains asserted. The $\overline{\text{FAIL}}$ pin is reasserted while the processor performs an external bus self-confidence test. If this self-test passes, the processor deasserts the $\overline{\text{FAIL}}$ pin and branches to the user's initialization routine; otherwise the $\overline{\text{FAIL}}$ pin remains asserted. Internal self-test and the use of the $\overline{\text{FAIL}}$ pin can be disabled with the STEST pin.
STEST	I S(L) H(Z) R(Z)	**SELF TEST** enables or disables the internal self-test feature at initialization. STEST is read on the rising edge of $\overline{\text{RESET}}$. When asserted, internal self-test and external bus confidence tests are performed during processor initialization. When deasserted, only the bus confidence tests are performed during initialization.
$\overline{\text{ONCE}}$	I A(L) H(Z) R(Z)	**ON CIRCUIT EMULATION**, when asserted, causes all outputs to be floated. $\overline{\text{ONCE}}$ is continuously sampled while $\overline{\text{RESET}}$ is low and is latched on the rising edge of $\overline{\text{RESET}}$. To place the processor in the ONCE state: (1) assert $\overline{\text{RESET}}$ and $\overline{\text{ONCE}}$ (order does not matter) (2) wait for at least 16 CLKIN periods in 2-x mode—or 10,000 CLKIN periods in 1-x mode—after V_{CC} and CLKIN are within operating specifications (3) deassert $\overline{\text{RESET}}$ (4) wait at least 32 CLKIN periods (The processor will now be latched in the ONCE state while $\overline{\text{RESET}}$ is high.) To exit the ONCE state, bring V_{CC} and CLKIN to operating conditions, then assert $\overline{\text{RESET}}$ and bring $\overline{\text{ONCE}}$ high prior to deasserting $\overline{\text{RESET}}$. CLKIN must operate within the specified operating conditions until Step 4 completes. CLKIN may then be changed to DC to achieve the lowest possible ONCE mode leakage current. $\overline{\text{ONCE}}$ can be used by emulator products or board testers to effectively make an installed processor transparent in the board.

	R(Z)	PCLK2:1 and the processor's internal clock. CLKMODE should be tied high or low in a system as the clock mode is not latched by the processor. If left unconnected, the processor internally pulls the CLKMODE pin low, enabling the 2-x clock mode.
PCLK2:1	O S H(Q) R(Q)	**PROCESSOR OUTPUT CLOCKS** provide a timing reference for all inputs and outputs. All input and output timings are specified in relation to PCLK2 and PCLK1. PCLK2 and PCLK1 are identical signals. Two output pins are provided to allow flexibility in the system's allocation of capacitive loading on the clock. PCLK2:1 may also be connected at the processor to form a single clock signal.
V_{SS}	–	**GROUND** connections must be connected externally to a V_{SS} board plane.
V_{CC}	–	**POWER** connections must be connected externally to a V_{CC} board plane.
V_{CCPLL}	–	**V_{CCPLL}** is a separate V_{CC} supply pin for the phase lock loop used in 1-x clock mode. Connecting a simple lowpass filter to V_{CCPLL} may help reduce clock jitter (T_{CP}) in noisy environments. Otherwise, V_{CCPLL} should be connected to V_{CC}.
NC	–	**NO CONNECT** pins must not be connected in a system.

Table 4. 80960CF Pin Description — DMA and Interrupt Unit Control Signals

Name	Type	Description
DREQ3:0	I A(L) H(Z) R(Z)	**DMA REQUEST** is used to request a DMA transfer. Each of the four signals requests a transfer on a single channel. DREQ0 requests channel 0, DREQ1 requests channel 1, etc. When two or more channels are requested simultaneously, the channel with the highest priority is serviced first. Channel priority mode is programmable.
DACK3:0	O S H(1) R(1)	**DMA ACKNOWLEDGE** indicates that a DMA transfer is being executed. Each of the four signals acknowledges a transfer for a single channel. DACK0 acknowledges channel 0, DACK1 acknowledges channel 1, etc. DACK3:0 are asserted when the requesting device of a DMA is accessed.
EOP/TC3:0	I/O A(L) H(Z/Q) R(Z)	**END OF PROCESS/TERMINAL COUNT** can be programmed as either an input (EOP3:0) or output (TC3:0), but not both. Each pin is individually programmable. When programmed as an input, EOPx causes termination of a current DMA transfer for the channel that corresponds to the EOPx pin. EOP0 corresponds to channel 0, EOP1 corresponds to channel 1, etc. When a channel is configured for source *and* destination chaining, the EOP pin for that channel causes termination of only the current buffer transferred and causes the next buffer to be transferred. EOP3:0 are asynchronous inputs. When programmed as an output, the channel's TCx pin indicates that the channel byte count has reached 0 and a DMA has terminated. TCx is driven with the same timing as DACKx during the last DMA transfer for a buffer. If the last bus request is executed as multiple bus accesses, TCx stays asserted for the entire bus request.
XINT7:0	I A(E/L) H(Z) R(Z)	**EXTERNAL INTERRUPT PINS** cause interrupts to be requested. These pins can be configured in three modes: **Dedicated Mode:** each pin is a dedicated external interrupt source. Dedicated inputs can be individually programmed to be level (low) or edge (falling) activated. **Expanded Mode:** the eight pins act together as an 8-bit vectored interrupt source. The interrupt pins in this mode are level activated. Since the interrupt pins are active low, the vector number requested is the 1's complement of the positive logic value place on the port. This eliminates glue logic to interface to combinational priority encoders which output negative logic. **Mixed Mode:** XINT7:5 are dedicated sources and XINT4:0 act as the five most significant bits of an expanded mode vector. The least significant bits are set to 010 internally.
NMI	I A(E) H(Z) R(Z)	**NON-MASKABLE INTERRUPT** causes a non-maskable interrupt event to occur. NMI is the highest priority interrupt recognized. NMI is an edge (falling) activated source.

3.3 80960CF Mechanical Data

3.3.1 80960CF PGA PINOUT

Figure 2 depicts the complete 80960CF PGA pinout as viewed from the top side of the component (i.e., pins facing down). Figure 3 shows the complete 80960CF PGA pinout as viewed from the pin-side of the package (i.e., pins facing up).

Table 5 lists the 80960CF pin names and package location in signal order; Table 6 lists the pin names and package location in pin order. See **Section 4.0, ELECTRICAL SPECIFICATIONS** for specifications and recommended connections.

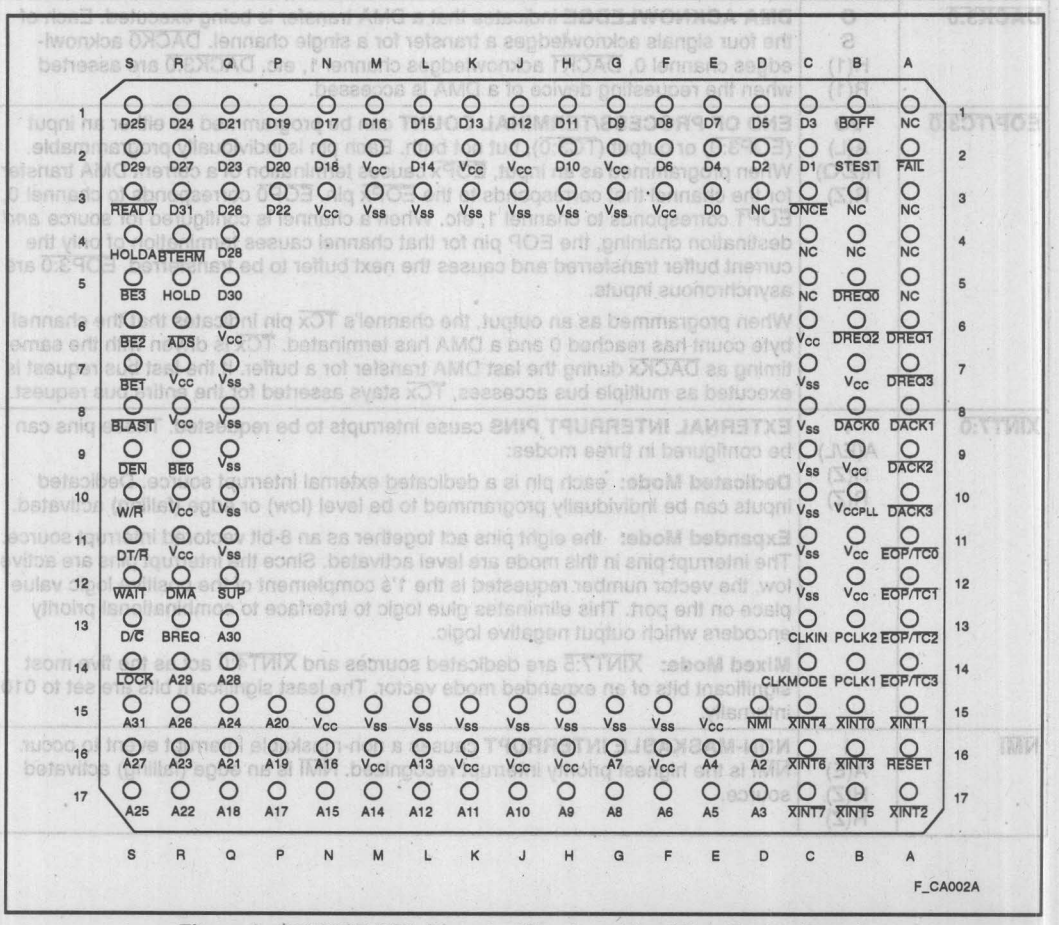


Figure 2. 80960CF PGA Pinout—View from Top (Pins Facing Down)

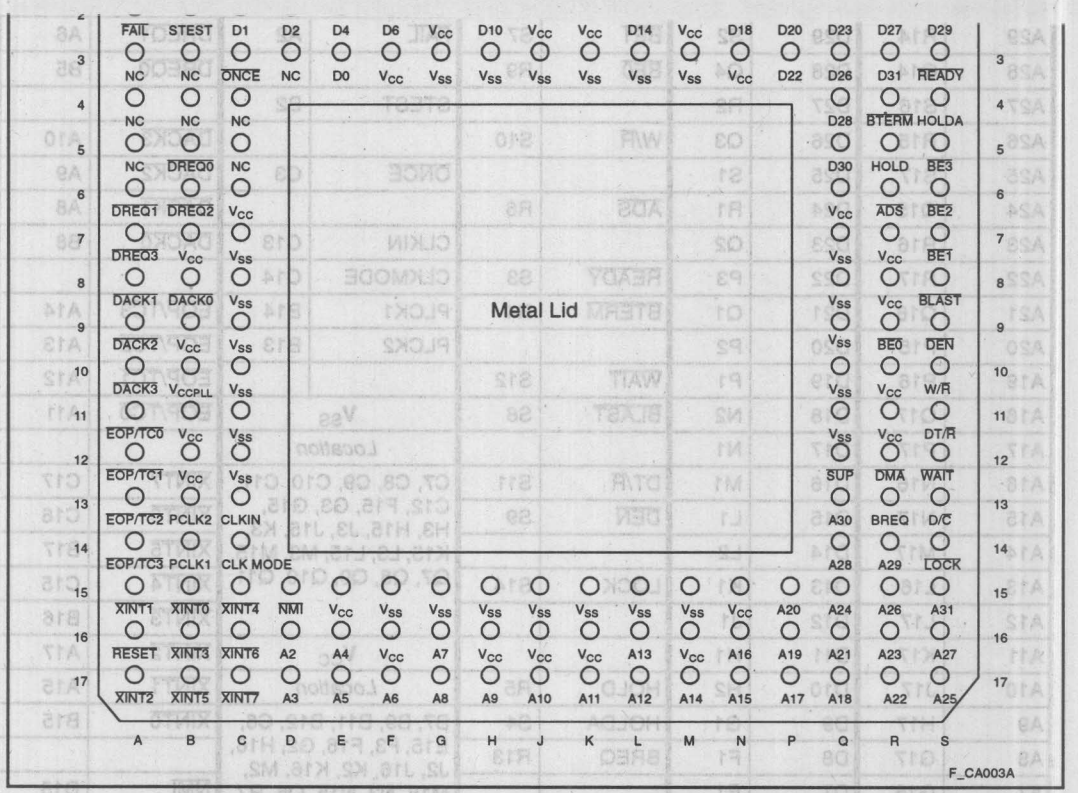


Figure 3. 80960CF PGA Pinout — View from Bottom (Pins Facing Up)

intel

Table 5. 80960CF PGA Pinout — In Signal Order

Address Bus		Data Bus		Bus Control		Processor Control		I/O	
Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin
A31	S15	D31	R3	$\overline{BE3}$	S5	$\overline{RESET}$	A16	$\overline{DREQ3}$	A7
A30	Q13	D30	Q5	$\overline{BE2}$	S6			$\overline{DREQ2}$	B6
A29	R14	D29	S2	$\overline{BE1}$	S7	$\overline{FAIL}$	A2	$\overline{DREQ1}$	A6
A28	Q14	D28	Q4	$\overline{BE0}$	R9			$\overline{DREQ0}$	B5
A27	S16	D27	R2			STEST	B2		
A26	R15	D26	Q3	W/$\overline{R}$	S10			$\overline{DACK3}$	A10
A25	S17	D25	S1			$\overline{ONCE}$	C3	$\overline{DACK2}$	A9
A24	Q15	D24	R1	$\overline{ADS}$	R6			$\overline{DACK1}$	A8
A23	R16	D23	Q2			CLKIN	C13	$\overline{DACK0}$	B8
A22	R17	D22	P3	$\overline{READY}$	S3	CLKMODE	C14		
A21	Q16	D21	Q1	$\overline{BTERM}$	R4	PLCK1	B14	$\overline{EOP/TC3}$	A14
A20	P15	D20	P2			PLCK2	B13	$\overline{EOP/TC2}$	A13
A19	P16	D19	P1	$\overline{WAIT}$	S12			$\overline{EOP/TC1}$	A12
A18	Q17	D18	N2	$\overline{BLAST}$	S8	V_{SS}		$\overline{EOP/TC0}$	A11
A17	P17	D17	N1			*Location*			
A16	N16	D16	M1	DT/$\overline{R}$	S11	C7, C8, C9, C10, C11, C12, F15, G3, G15, H3, H15, J3, J15, K3, K15, L3, L15, M3, M15, Q7, Q8, Q9, Q10, Q11		$\overline{XINT7}$	C17
A15	N17	D15	L1	$\overline{DEN}$	S9			$\overline{XINT6}$	C16
A14	M17	D14	L2					$\overline{XINT5}$	B17
A13	L16	D13	K1	LOCK	S14			$\overline{XINT4}$	C15
A12	L17	D12	J1					$\overline{XINT3}$	B16
A11	K17	D11	H1			V_{CC}		$\overline{XINT2}$	A17
A10	J17	D10	H2	HOLD	R5	*Location*		$\overline{XINT1}$	A15
A9	H17	D9	G1	HOLDA	S4	B7, B9, B11, B12, C6, E15, F3, F16, G2, H16, J2, J16, K2, K16, M2, M16, N3, N15, Q6, R7, R8, R10, R11		$\overline{XINT0}$	B15
A8	G17	D8	F1	BREQ	R13				
A7	G16	D7	E1					$\overline{NMI}$	D15
A6	F17	D6	F2	D/$\overline{C}$	S13				
A5	E17	D5	D1	$\overline{DMA}$	R12				
A4	E16	D4	E2	$\overline{SUP}$	Q12	V_{CCPLL}	B10		
A3	D17	D3	C1			No Connect			
A2	D16	D2	D2	$\overline{BOFF}$	B1	*Location*			
		D1	C2			A1, A3, A4, A5, B3, B4, C4, C5, D3			
		D0	E3						

Table 6. 80960CF PGA Pinout — In Pin Order

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
A1	NC	C1	D3	F17	A6	M15	V_{SS}	R3	D31
A2	$\overline{FAIL}$	C2	D1	G1	D9	M16	V_{CC}	R4	$\overline{BTERM}$
A3	NC	C3	$\overline{ONCE}$	G2	V_{CC}	M17	A14	R5	HOLD
A4	NC	C4	NC	G3	V_{SS}	N1	D17	R6	$\overline{ADS}$
A5	NC	C5	NC	G15	V_{SS}	N2	D18	R7	V_{CC}
A6	$\overline{DREQ1}$	C6	V_{CC}	G16	A7	N3	V_{CC}	R8	V_{CC}
A7	$\overline{DREQ3}$	C7	V_{SS}	G17	A8	N15	V_{CC}	R9	$\overline{BE0}$
A8	$\overline{DACK1}$	C8	V_{SS}	H1	D11	N16	A16	R10	V_{CC}
A9	$\overline{DACK2}$	C9	V_{SS}	H2	D10	N17	A15	R11	V_{CC}
A10	$\overline{DACK3}$	C10	V_{SS}	H3	V_{SS}	P1	D19	R12	$\overline{DMA}$
A11	$\overline{EOP/TC0}$	C11	V_{SS}	H15	V_{SS}	P2	D20	R13	BREQ
A12	$\overline{EOP/TC1}$	C12	V_{SS}	H16	V_{CC}	P3	D22	R14	A29
A13	$\overline{EOP/TC2}$	C13	CLKIN	H17	A9	P15	A20	R15	A26
A14	$\overline{EOP/TC3}$	C14	CLKMODE	J1	D12	P16	A19	R16	A23
A15	$\overline{XINT1}$	C15	$\overline{XINT4}$	J2	V_{CC}	P17	A17	R17	A22
A16	$\overline{RESET}$	C16	$\overline{XINT6}$	J3	V_{SS}	Q1	D21	S1	D25
A17	$\overline{XINT2}$	C17	$\overline{XINT7}$	J15	V_{SS}	Q2	D23	S2	D29
B1	$\overline{BOFF}$	D1	D5	J16	V_{CC}	Q3	D26	S3	$\overline{READY}$
B2	STEST	D2	D2	J17	A10	Q4	D28	S4	HOLDA
B3	NC	D3	NC	K1	D13	Q5	D30	S5	$\overline{BE3}$
B4	NC	D15	$\overline{NMI}$	K2	V_{CC}	Q6	V_{CC}	S6	$\overline{BE2}$
B5	$\overline{DREQ0}$	D16	A2	K3	V_{SS}	Q7	V_{SS}	S7	$\overline{BE1}$
B6	$\overline{DREQ2}$	D17	A3	K15	V_{SS}	Q8	V_{SS}	S8	$\overline{BLAST}$
B7	V_{CC}	E1	D7	K16	V_{CC}	Q9	V_{SS}	S9	$\overline{DEN}$
B8	$\overline{DACK0}$	E2	D4	K17	A11	Q10	V_{SS}	S10	W/$\overline{R}$
B9	V_{CC}	E3	D0	L1	D15	Q11	V_{SS}	S11	DT/$\overline{R}$
B10	V_{CCPLL}	E15	V_{CC}	L2	D14	Q12	$\overline{SUP}$	S12	$\overline{WAIT}$
B11	V_{CC}	E16	A4	L3	V_{SS}	Q13	A30	S13	D/$\overline{C}$
B12	V_{CC}	E17	A5	L15	V_{SS}	Q14	A28	S14	$\overline{LOCK}$
B13	PCLK2	F1	D8	L16	A13	Q15	A24	S15	A31
B14	PCLK1	F2	D6	L17	A12	Q16	A21	S16	A27
B15	$\overline{XINT0}$	F3	V_{CC}	M1	D16	Q17	A18	S17	A25
B16	$\overline{XINT3}$	F15	V_{SS}	M2	V_{CC}	R1	D24		
B17	$\overline{XINT5}$	F16	V_{CC}	M3	V_{SS}	R2	D27		

Table 7. 80960CF PQFP Pinout — In Signal Order (80960CF-33, -25, -16 Only)

Address Bus		Data Bus		Bus Control		Processor Control		I/O	
Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin
A31	153	D31	186	$\overline{BE3}$	176	$\overline{RESET}$	91	$\overline{DREQ3}$	60
A30	152	D30	187	$\overline{BE2}$	175	$\overline{FAIL}$	45	$\overline{DREQ2}$	59
A29	151	D29	188	$\overline{BE1}$	172	STEST	46	$\overline{DREQ1}$	58
A28	145	D28	189	$\overline{BE0}$	170	$\overline{ONCE}$	43	$\overline{DREQ0}$	57
A27	144	D27	191			CLKIN	87		
A26	143	D26	192	W/$\overline{R}$	164	CLKMODE	85	$\overline{DACK3}$	65
A25	142	D25	194			PCLK2	74	$\overline{DACK2}$	64
A24	141	D24	195	$\overline{ADS}$	178	PCLK1	78	$\overline{DACK1}$	63
A23	139	D23	3			V_{SS}		$\overline{DACK0}$	62
A22	138	D22	4	$\overline{READY}$	182	*Location*			
A21	137	D21	5	$\overline{BTERM}$	184	2, 7, 16, 24, 30, 38, 39, 49, 56, 70, 75, 77, 81, 83, 88, 89, 92, 98, 105, 109, 110, 121, 125, 131, 135, 147, 150, 161, 165, 173, 174, 185, 196		$\overline{EOP/TC3}$	69
A20	136	D20	6					$\overline{EOP/TC2}$	68
A19	134	D19	8	$\overline{WAIT}$	162			$\overline{EOP/TC1}$	67
A18	133	D18	9	$\overline{BLAST}$	169			$\overline{EOP/TC0}$	66
A17	132	D17	10						
A16	130	D16	11	DT/$\overline{R}$	163			$\overline{XINT7}$	107
A15	129	D15	13	$\overline{DEN}$	167	V_{CC}		$\overline{XINT6}$	106
A14	128	D14	14			*Location*		$\overline{XINT5}$	102
A13	124	D13	15	$\overline{LOCK}$	156	1, 12, 20, 28, 32, 37, 44, 50, 61, 71, 79, 82, 96, 99, 103, 115, 127, 140, 148, 154, 168, 171, 180, 190		$\overline{XINT4}$	101
A12	123	D12	17					$\overline{XINT3}$	100
A11	122	D11	18	HOLD	181			$\overline{XINT2}$	95
A10	120	D10	19	HOLDA	179			$\overline{XINT1}$	94
A9	119	D9	21	BREQ	155			$\overline{XINT0}$	93
A8	118	D8	22			V_{CCPLL}	72		
A7	117	D7	23	D/$\overline{C}$	159	No Connect		$\overline{NMI}$	108
A6	116	D6	25	$\overline{DMA}$	160	*Location*			
A5	114	D5	26	$\overline{SUP}$	158	29, 31, 41, 42, 47, 48, 51, 52, 53, 54, 55, 73, 76, 80, 84, 86, 90, 97, 104, 126, 146, 149, 157, 166, 177, 183, 193			
A4	113	D4	27						
A3	112	D3	33	$\overline{BOFF}$	40				
A2	111	D2	34						
		D1	35						
		D0	36						

Table 8. 80960CF PQFP Pinout — In Pin Order (80960CF-33, -25, -16 Only)

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
1	V_{CC}	34	D2	67	EOP/TC1	100	XINT3	133	A18	166	NC
2	V_{SS}	35	D1	68	EOP/TC2	101	XINT4	134	A19	167	DEN
3	D23	36	D0	69	EOP/TC3	102	XINT5	135	V_{SS}	168	V_{CC}
4	D22	37	V_{CC}	70	V_{SS}	103	V_{CC}	136	A20	169	BLAST
5	D21	38	V_{SS}	71	V_{CC}	104	NC	137	A21	170	BE0
6	D20	39	V_{SS}	72	V_{CCPLL}	105	V_{SS}	138	A22	171	V_{CC}
7	V_{SS}	40	BOFF	73	NC	106	XINT6	139	A23	172	BE1
8	D19	41	NC	74	PCLK2	107	XINT7	140	V_{CC}	173	V_{SS}
9	D18	42	NC	75	V_{SS}	108	NMI	141	A24	174	V_{SS}
10	D17	43	ONCE	76	NC	109	V_{SS}	142	A25	175	BE2
11	D16	44	V_{CC}	77	V_{SS}	110	V_{SS}	143	A26	176	BE3
12	V_{CC}	45	FAIL	78	PCLK1	111	A2	144	A27	177	NC
13	D15	46	STEST	79	V_{CC}	112	A3	145	A28	178	ADS
14	D14	47	NC	80	NC	113	A4	146	NC	179	HOLDA
15	D13	48	NC	81	V_{SS}	114	A5	147	V_{SS}	180	V_{CC}
16	V_{SS}	49	V_{SS}	82	V_{CC}	115	V_{CC}	148	V_{CC}	181	HOLD
17	D12	50	V_{CC}	83	V_{SS}	116	A6	149	NC	182	READY
18	D11	51	NC	84	NC	117	A7	150	V_{SS}	183	NC
19	D10	52	NC	85	CLKMODE	118	A8	151	A29	184	BTERM
20	V_{CC}	53	NC	86	NC	119	A9	152	A30	185	V_{SS}
21	D9	54	NC	87	CLKIN	120	A10	153	A31	186	D31
22	D8	55	NC	88	V_{SS}	121	V_{SS}	154	V_{CC}	187	D30
23	D7	56	V_{SS}	89	V_{SS}	122	A11	155	BREQ	188	D29
24	V_{SS}	57	DREQ0	90	NC	123	A12	156	LOCK	189	D28
25	D6	58	DREQ1	91	RESET	124	A13	157	NC	190	V_{CC}
26	D5	59	DREQ2	92	V_{SS}	125	V_{SS}	158	SUP	191	D27
27	D4	60	DREQ3	93	XINT0	126	NC	159	D/C	192	D26
28	V_{CC}	61	V_{CC}	94	XINT1	127	V_{CC}	160	DMA	193	NC
29	NC	62	DACK0	95	XINT2	128	A14	161	V_{SS}	194	D25
30	V_{SS}	63	DACK1	96	V_{CC}	129	A15	162	WAIT	195	D24
31	NC	64	DACK2	97	NC	130	A16	163	DT/R	196	V_{SS}
32	V_{CC}	65	DACK3	98	V_{SS}	131	V_{SS}	164	W/R		
33	D3	66	EOP/TC0	99	V_{CC}	132	A17	165	V_{SS}		

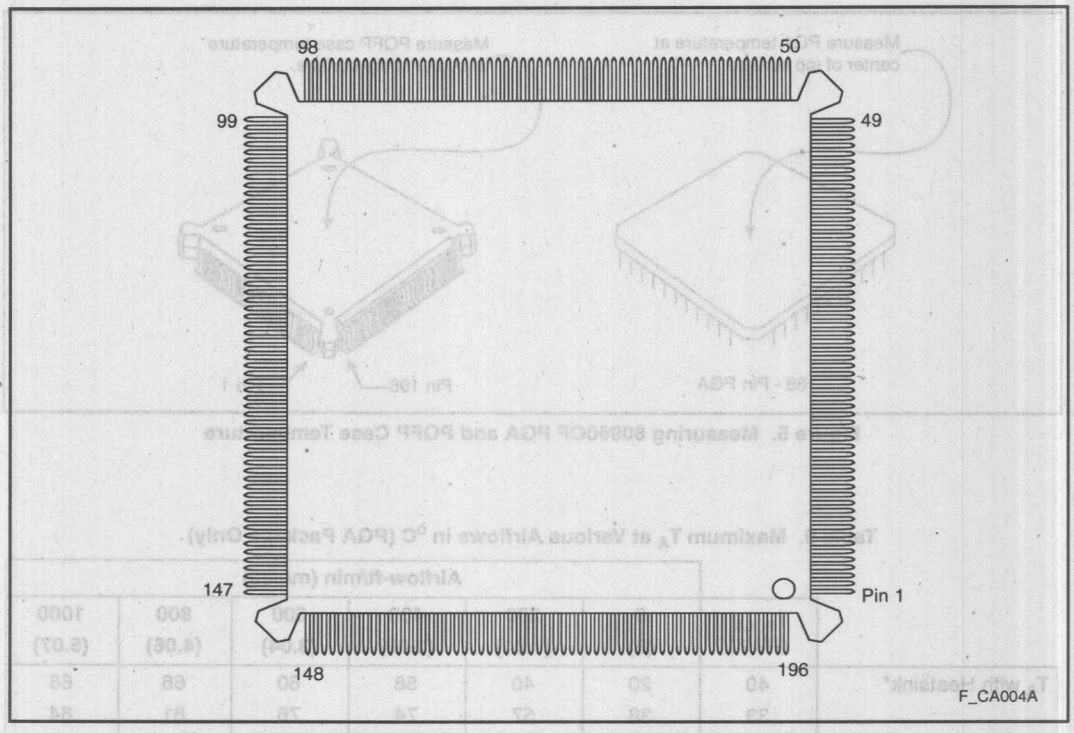


Figure 4. 80960CF PQFP Pinout—Top View (80960CF-33, -25, -16 Only)

3.4 Package Thermal Specifications

The 80960CF is specified for operation when T_C (case temperature) is within the range of 0°C–100°C for 33, 25, and 16 MHz and 0°C–85°C for 40 MHz. T_C may be measured in any environment to determine whether the 80960CF is within specified operating range. Case temperature should be measured at the center of the top surface, opposite the pins. Refer to Figure 5.

T_A (ambient temperature) is calculated from θ_{CA} (thermal resistance from case to ambient) using the equation:

$$T_A = T_C - P^*\theta_{CA}$$

Table 9 shows the maximum T_A allowable (without exceeding T_C) at various airflows and operating frequencies (f_{PCLK}).

Note that T_A is greatly improved by attaching fins or a heatsink to the package. P (maximum power consumption) is calculated by using the typical I_{CC} as tabulated in **Section 4.4, DC Specifications** and V_{CC} of 5 V.

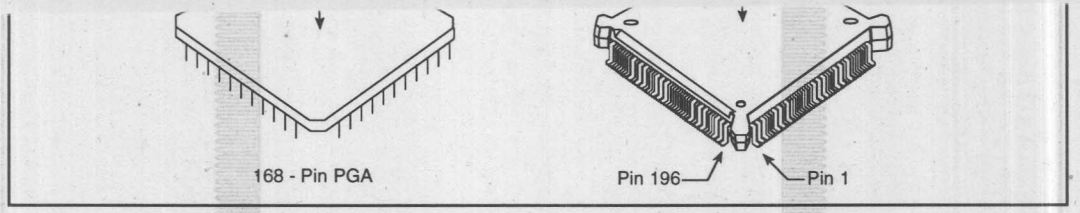


Figure 5. Measuring 80960CF PGA and PQFP Case Temperature

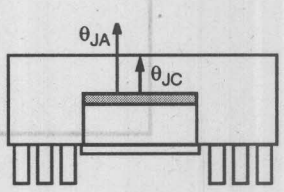
Table 9. Maximum T_A at Various Airflows in °C (PGA Package Only)

		Airflow-ft/min (m/sec)					
	f_{PCLK} (MHz)	0 (0)	200 (1.01)	400 (2.03)	600 (3.04)	800 (4.06)	1000 (5.07)
T_A with Heatsink*	40	20	40	58	60	66	68
	33	38	57	74	76	81	84
	25	50	65	79	81	85	87
	16	63	74	84	86	89	90
T_A without Heatsink*	40	0	15	30	40	50	52
	33	18	33	47	57	66	67
	25	34	46	57	65	72	74
	16	51	60	68	74	80	81

NOTES:
*0.285" high unidirectional heatsink (Al alloy 6061, 50 mil fin width, 150 mil center-to-center fin spacing).

Table 10. 80960CF PGA Package Thermal Characteristics

Thermal Resistance — °C/Watt						
Parameter	Airflow — ft./min (m/sec)					
	0 (0)	200 (1.01)	400 (2.03)	600 (3.07)	800 (4.06)	1000 (5.07)
θ Junction-to-Case (Case measured as shown in Figure 5)	1.5	1.5	1.5	1.5	1.5	1.5
θ Case-to-Ambient (No Heatsink)	17	14	11	9	7.1	6.6
θ Case-to-Ambient (With Heatsink)*	13	9	5.5	5	3.9	3.4



NOTES:

1. This table applies to 80960CF PGA plugged into socket or soldered directly to board.
2. $\theta_{JA} = \theta_{JC} + \theta_{CA}$

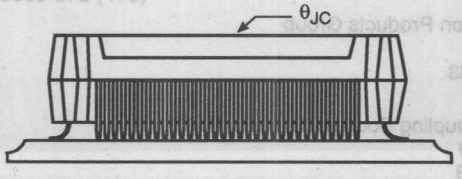
*0.285" high unidirectional heatsink (Al alloy 6061, 50 mil fin width, 150 mil center-to-center fin spacing).

Table 11. 80960CF PQFP Package Thermal Characteristics

Thermal Resistance — °C/Watt							
Parameter	Airflow — ft./min (m/sec)						
	0 (0)	50 (0.25)	100 (0.50)	200 (1.01)	400 (2.03)	600 (3.04)	800 (4.06)
θ Junction-to-Case (Case Measured as shown in Figure 5)	5	5	5	5	5	5	5
θ Case-to-Ambient (No Heatsink)	19	18	17	15	12	10	9

NOTES:

1. This table applies to 80960CF PQFP soldered directly to board.
2. $\theta_{JA} = \theta_{JC} + \theta_{CA}$



3.5 Stepping Register Information

Upon reset, register g0 contains die stepping information (Figure 6). The most significant byte contains ASCII 0; the upper middle byte contains an ASCII C; the lower middle byte contains an ASCII F. The least significant byte contains the stepping number in ASCII. g0 retains this information until it is overwritten by the user program. Table 12 contains a cross reference of the number in the least significant byte of register g0 to the die stepping number.

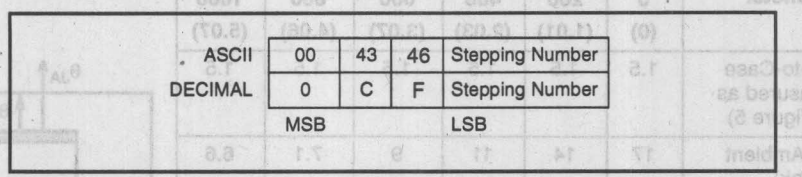


Figure 6. Register g0

Table 12. Die Stepping Cross Reference

g0 Least Significant Byte	Die Stepping
01	A
02	B
03	C
04	D
05	E

3.6 Sources for Accessories

The following is a list of suggested sources for 80960CF accessories. This is neither an endorsement nor a warranty of the performance of any of the listed products and/or companies.

Sockets

1. 3M Textool Test and Interconnection Products
   6801 River Place Blvd. Mailstop 130-3N-29
   Austin, TX 78726-9000
   (800) 328-0411
2. Augat, Inc. Interconnection Products Group
   452 John Dietsch Blvd.
   Attleboro Falls, MA 02763
   (508) 699-7646
3. Concept Mfg., Inc. (Decoupling Sockets)
   400 Walnut St. Suite 609
   Redwood City, CA 94063
   (415) 365-1162 FAX: (415)265-1164

Heatsinks/Fins

1. Thermalloy, Inc.
   2021 West Valley View Lane
   Dallas, TX 75234
   (214) 243-4321
   FAX: (214) 241-4656
2. Wakefield Engineering
   60 Audubon Road
   Wakefield, MA 01880
   (617) 245-5900

4.0 ELECTRICAL SPECIFICATIONS

4.1 Absolute Maximum Ratings

Parameter	Maximum Rating
Storage Temperature	–65°C to +150°C
Case Temperature Under Bias	–65°C to +110°C
Supply Voltage wrt. V_{SS}	–0.5 V to + 6.5 V
Voltage on Other Pins wrt. V_{SS}	–0.5 V to V_{CC} + 0.5 V

NOTICE: This document contains preliminary information on new products in production.The specifications are subject to change without notice. Verify with your local Intel sales office that you have the latest datasheet before finalizing a design.

***WARNING:** *Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

4.2 Operating Conditions

Table 13. Operating Conditions

Symbol	Parameter	Min	Max	Units	Notes
V_{CC}	Supply Voltage				
	80960CF-40	4.75	5.25	V	
	80960CF-33	4.50	5.50		
	80960CF-25	4.50	5.50		
	80960CF-16	4.50	5.50		
f_{CLK2x}	Input Clock Frequency (2-x Mode)				
	80960CF-40	0	80	MHz	
	80960CF-33	0	66		
	80960CF-25	0	50		
	80960CF-16	0	32		
f_{CLK1x}	Input Clock Frequency (1-x Mode)				(1)
	80960CF-40	8	40	MHz	
	80960CF-33	8	33		
	80960CF-25	8	25		
	80960CF-16	8	16		
T_C	Case Temp Under Bias				
	PGA Pkg. (80960CF-40)	0	85	°C	
	PGA Pkg. (80960CF-33, -25, -16 Only)	0	100		
	196-Pin PQFP (80960CF-33, -25, -16 Only)	0	100		

NOTES:

1. When in the 1-x input clock mode, CLKIN is an input to an internal phase-locked loop and must maintain a minimum frequency of 8 MHz for proper processor operation. However, in the 1-x mode, CLKIN may still be stopped when the processor is in a reset condition. If CLKIN is stopped, the specified $\overline{RESET}$ low time must be provided once CLKIN restarts and has stabilized.

Liberal decoupling capacitance should be placed near the 80960CF. The processor can cause transient power surges when its numerous output buffers transition, particularly when connected to large capacitive loads.

Low inductance capacitors and interconnects are recommended for best high frequency electrical pull-up resistor. Pull-up resistors should be in the in the range of 20 KΩ for each pin tied high. If $\overline{\text{READY}}$ or HOLD are not used, the unused input should be connected to ground. **N.C. pins must always remain unconnected.** For additional information refer to the *i960® Cx Microprocessor User's Manual* (270710).

4.4 DC Specifications

Table 14. DC Characteristics (Sheet 1 of 2)

(80960CF-40, 33, -25, -16 under the conditions described in **Section 4.2, Operating Conditions**.)

Symbol	Parameter		Min	Max	Units	Notes
V_{IL}	Input Low Voltage for all pins except $\overline{\text{RESET}}$		– 0.3	+0.8	V	
V_{IH}	Input High Voltage for all pins except $\overline{\text{RESET}}$		2.0	V_{CC} + 0.3	V	
V_{OL}	Output Low Voltage			0.45	V	I_{OL} = 5 mA
V_{OH}	Output High Voltage	I_{OH} = –1 mA I_{OH} = – 200 µA	2.4 V_{CC} – 0.5		V V	
V_{ILR}	Input Low Voltage for $\overline{\text{RESET}}$		– 0.3	1.5	V	
V_{IHR}	Input High Voltage for $\overline{\text{RESET}}$		3.5	V_{CC} + 0.3	V	

NOTES:

1. No pullup or pulldown.
2. These pins have internal pullup resistors.
3. These pins have internal pulldown resistors.
4. Measured at worst case frequency, V_{CC} and temperature, with device operating and outputs loaded to the test conditions described in **Section 4.5.1, AC TEST CONDITIONS.**
5. I_{CC} Typical is not tested.
6. Output Capacitance is the capacitive load of a floating output.
7. CLKMODE pin has a pulldown resistor only when $\overline{\text{ONCE}}$ pin is deasserted.

Table 14. DC Characteristics (Sheet 2 of 2)

(80960CF-40, 33, -25, -16 under the conditions described in **Section 4.2, Operating Conditions**.)

Symbol	Parameter	Min	Max	Units	Notes
I_{LI1}	Input Leakage Current for each pin *except*: BTERM, ONCE, DREQ3:0, STEST, EOP3:0/TC3:0, NMI, XINT7:0, BOFF, READY, HOLD, CLKMODE		±15	µA	$0 \le V_{IN} \le V_{CC}$ (1)
I_{LI2}	Input Leakage Current for: BTERM, ONCE, DREQ3:0, STEST, EOP3:0/TC3:0, NMI, XINT7:0, BOFF	0	– 300	µA	V_{IN} = 0.45 V (2)
I_{LI3}	Input Leakage Current for: READY, HOLD, CLKMODE	0	500	µA	V_{IN} = 2.4 V (3,7)
I_{LO}	Output Leakage Current		±15	µA	$0.45 \le V_{OUT} \le V_{CC}$
I_{CC}	Supply Current (80960CF-40, 33):				
	I_{CC} Max		1150	mA	(4)
	I_{CC} Typ		1000	mA	(5)
I_{CC}	Supply Current (80960CF-25):				
	I_{CC} Max		950	mA	(4)
	I_{CC} Typ		775	mA	(5)
I_{CC}	Supply Current (80960CF-16):			mA	
	I_{CC} Max		750		(4)
	I_{CC} Typ		575		(5)
I_{ONCE}	ONCE-mode Supply Current			mA	
	80960CF-40		225		
	80960CF-33, -25, -16		150		
C_{IN}	Input Capacitance for: CLKIN, RESET, ONCE, READY, HOLD, DREQ3:0, BOFF, XINT7:0, NMI, BTERM, CLKMODE	0	12	pF	F_C = 1 MHz
C_{OUT}	Output Capacitance of each output pin		12	pF	F_C = 1 MHz (6)
$C_{I/O}$	I/O Pin Capacitance		12	pF	F_C = 1 MHz

NOTES:

1. No pullup or pulldown.
2. These pins have internal pullup resistors.
3. These pins have internal pulldown resistors.
4. Measured at worst case frequency, V_{CC} and temperature, with device operating and outputs loaded to the test conditions described in **Section 4.5.1, AC TEST CONDITIONS.**
5. I_{CC} Typical is not tested.
6. Output Capacitance is the capacitive load of a floating output.
7. CLKMODE pin has a pulldown resistor only when ONCE pin is deasserted.

4.5 AC Specifications

Table 15. 80960CF AC Characteristics (40 MHz) (Sheet 1 of 3)

(80960CF-40 only, per the conditions in **4.2 Operating Conditions** and **4.5.1 AC TEST CONDITIONS**.)

Symbol	Parameter		Min	Max	Units	Notes
Input Clock (1,9)						
T_F	CLKIN Frequency		0	80	MHz	
T_C	CLKIN Period	In 1-x Mode (f_{CLK1x}) In 2-x Mode (f_{CLK2x})	25 12.5	125 ∞	ns ns	(11)
T_{CS}	CLKIN Period Stability	In 1-x Mode (f_{CLK1x})		±0.1%	Δ	(12)
T_{CH}	CLKIN High Time	In 1-x Mode (f_{CLK1x}) In 2-x Mode (f_{CLK2x})	5 5	62.5 ∞	ns ns	(11)
T_{CL}	CLKIN Low Time	In 1-x Mode (f_{CLK1x}) In 2-x Mode (f_{CLK2x})	5 5	62.5 ∞	ns ns	(11)
T_{CR}	CLKIN Rise Time		0	6	ns	
T_{CF}	CLKIN Fall Time		0	6	ns	
Output Clocks (1,8)						
T_{CP}	CLKIN to PCLK2:1 Delay	In 1-x Mode (f_{CLK1x}) In 2-x Mode (f_{CLK2x})	– 2 2	2 25	ns ns	(3,12) (3)
T	PCLK2:1 Period	In 1-x Mode (f_{CLK1x}) In 2-x Mode (f_{CLK2x})	T_C $2T_C$		ns ns	(12) (3)
T_{PH}	PCLK2:1 High Time		(T/2) – 2	T/2	ns	(12)
T_{PL}	PCLK2:1 Low Time		(T/2) – 2	T/2	ns	(12)
T_{PR}	PCLK2:1 Rise Time		1	4	ns	(3)
T_{PF}	PCLK2:1 Fall Time		1	4	ns	(3)
Synchronous Outputs (8)						

NOTES:
See Table 18 for all notes related to AC specifications.

Table 15. 80960CF AC Characteristics (40 MHz) (Sheet 2 of 3)

(80960CF-40 only, per the conditions in **4.2 Operating Conditions** and **4.5.1 AC TEST CONDITIONS.**)

Symbol	Parameter		Min	Max	Units	Notes
T_{OH}	Output Valid Delay, Output Hold					(6,10)
T_{OV}	T_{OH1}, T_{OV1}	A31:2	3	14	ns	
	T_{OH2}, T_{OV2}	$\overline{BE3:0}$	3	16	ns	
	T_{OH3}, T_{OV3}	$\overline{ADS}$	6	16	ns	
	T_{OH4}, T_{OV4}	W/$\overline{R}$	3	16	ns	
	T_{OH5}, T_{OV5}	D/$\overline{C}$, $\overline{SUP}$, $\overline{DMA}$	4	16	ns	
	T_{OH6}, T_{OV6}	$\overline{BLAST}$, $\overline{WAIT}$	5	16	ns	
	T_{OH7}, T_{OV7}	$\overline{DEN}$	3	16	ns	
	T_{OH8}, T_{OV8}	HOLDA, BREQ	4	16	ns	
	T_{OH9}, T_{OV9}	$\overline{LOCK}$	4	16	ns	
	T_{OH10}, T_{OV10}	$\overline{DACK3:0}$	4	16	ns	
	T_{OH11}, T_{OV11}	D31:0	3	16	ns	
	T_{OH12}, T_{OV12}	DT/$\overline{R}$	T/2 + 3	T/2 + 14	ns	
	T_{OH13}, T_{OV13}	$\overline{FAIL}$	2	14	ns	
	T_{OH14}, T_{OV14}	$\overline{EOP3:0}$/TC3:0	3	16	ns	(6,10)
T_{OF}	Output Float for all outputs		3	22	ns	(6)
Synchronous Inputs (1,9,10)						
T_{IS}	Input Setup					
	T_{IS1}	D31:0	3		ns	
	T_{IS2}	$\overline{BOFF}$	15		ns	
	T_{IS3}	$\overline{BTERM}$/READY	7		ns	
	T_{IS4}	HOLD	5		ns	
T_{IH}	Input Hold					
	T_{IH1}	D31:0	5		ns	
	T_{IH2}	$\overline{BOFF}$	5		ns	
	T_{IH3}	$\overline{BTERM}$/READY	2		ns	
	T_{IH4}	HOLD	3		ns	

NOTES:

See Table 18 for all notes related to AC specifications.

	Rising				
T_{AVEL1}	A31:2 Valid to $\overline{DEN}$ Falling	T – 4	T + 4	ns	
T_{AVEL2}	$\overline{BE3:0}$, W/$\overline{R}$, $\overline{SUP}$, $\overline{INST}$,$\overline{DMA}$, $\overline{DACK3:0}$ Valid to $\overline{DEN}$ Falling	T – 6	T + 6	ns	
T_{NLQV}	$\overline{WAIT}$ Falling to Output Data Valid	± 6		ns	
T_{DVNH}	Output Data Valid to $\overline{WAIT}$ Rising	N*T – 6	N*T + 6	ns	(4)
T_{NLNH}	$\overline{WAIT}$ Falling to $\overline{WAIT}$ Rising	N*T ± 4		ns	(4)
T_{NHQX}	Output Data Hold after $\overline{WAIT}$ Rising	(N+1)*T–8	(N+1)*T+6	ns	(5)
T_{EHTV}	DT/$\overline{R}$ Hold after $\overline{DEN}$ High	T/2 – 7	∞	ns	(6)
T_{TVEL}	DT/$\overline{R}$ Valid to $\overline{DEN}$ Falling	T/2 – 4		ns	
Relative Input Timings (1,2,3)					
T_{IS5}	$\overline{RESET}$ Input Setup (2-x Clock Mode)	6		ns	(13)
T_{IH5}	$\overline{RESET}$ Input Hold (2-x Clock Mode)	5		ns	(13)
T_{IS6}	$\overline{DREQ3:0}$ Input Setup	12		ns	(7)
T_{IH6}	$\overline{DREQ3:0}$ Input Hold	7		ns	(7)
T_{IS7}	$\overline{XINT7:0}$, NMI Input Setup	7		ns	(15)
T_{IH7}	$\overline{XINT7:0}$, NMI Input Hold	3		ns	(15)
T_{IS8}	$\overline{RESET}$ Input Setup (1-x Clock Mode)	3		ns	(14)
T_{IH8}	$\overline{RESET}$ Input Hold (1-x Clock Mode)	T/4 + 1		ns	(14)

NOTES:
See Table 18 for all notes related to AC specifications.

Table 16. 80960CF AC Characteristics (33 MHz) (Sheet 1 of 3)

(80960CF-33 only, per the conditions in **4.2 Operating Conditions** and **4.5.1 AC TEST CONDITIONS**.)

Symbol	Parameter		Min	Max	Units	Notes
Input Clock (1,9)						
T_F	CLKIN Frequency		0	66.66	MHz	
T_C	CLKIN Period	In 1-x Mode (f_{CLK1x}) In 2-x Mode (f_{CLK2x})	30 15	125 ∞	ns ns	(11)
T_{CS}	CLKIN Period Stability	In 1-x Mode (f_{CLK1x})		±0.1%	Δ	(12)
T_{CH}	CLKIN High Time	In 1-x Mode (f_{CLK1x}) In 2-x Mode (f_{CLK2x})	5 5	62.5 ∞	ns ns	(11)
T_{CL}	CLKIN Low Time	In 1-x Mode (f_{CLK1x}) In 2-x Mode (f_{CLK2x})	5 5	62.5 ∞	ns ns	(11)
T_{CR}	CLKIN Rise Time		0	6	ns	
T_{CF}	CLKIN Fall Time		0	6	ns	
Output Clocks (1,8)						
T_{CP}	CLKIN to PCLK2:1 Delay	In 1-x Mode (f_{CLK1x}) In 2-x Mode (f_{CLK2x})	– 2 2	2 25	ns ns	(3,12) (3)
T	PCLK2:1 Period	In 1-x Mode (f_{CLK1x}) In 2-x Mode (f_{CLK2x})	TC 2TC		ns ns	(12) (3)
T_{PH}	PCLK2:1 High Time		(T/2) – 2	T/2	ns	(12)
T_{PL}	PCLK2:1 Low Time		(T/2) – 2	T/2	ns	(12)
T_{PR}	PCLK2:1 Rise Time		1	4	ns	(3)
T_{PF}	PCLK2:1 Fall Time		1	4	ns	(3)

NOTES:
See Table 18 for all notes related to AC specifications.

Table 16. 80960CF AC Characteristics (33 MHz) (Sheet 2 of 3)

(80960CF-33 only, per the conditions in **4.2 Operating Conditions** and **4.5.1 AC TEST CONDITIONS**.)

Symbol	Parameter		Min	Max	Units	Notes
Synchronous Outputs (8)						
T_{OH} T_{OV}	Output Valid Delay, Output Hold					(6,10)
	T_{OH1}, T_{OV1}	A31:2	3	14	ns	
	T_{OH2}, T_{OV2}	$\overline{BE3:0}$	3	16	ns	
	T_{OH3}, T_{OV3}	$\overline{ADS}$	6	18	ns	
	T_{OH4}, T_{OV4}	W/$\overline{R}$	3	18	ns	
	T_{OH5}, T_{OV5}	D/$\overline{C}$, $\overline{SUP}$, $\overline{DMA}$	4	16	ns	
	T_{OH6}, T_{OV6}	$\overline{BLAST}$, $\overline{WAIT}$	5	16	ns	
	T_{OH7}, T_{OV7}	$\overline{DEN}$	3	16	ns	
	T_{OH8}, T_{OV8}	HOLDA, BREQ	4	16	ns	
	T_{OH9}, T_{OV9}	$\overline{LOCK}$	4	16	ns	
	T_{OH10}, T_{OV10}	$\overline{DACK3:0}$	4	18	ns	
	T_{OH11}, T_{OV11}	D31:0	3	16	ns	
	T_{OH12}, T_{OV12}	DT/$\overline{R}$	T/2 + 3	T/2 + 14	ns	
	T_{OH13}, T_{OV13}	$\overline{FAIL}$	2	14	ns	
	T_{OH14}, T_{OV14}	$\overline{EOP3:0}$/$\overline{TC3:0}$	3	18	ns	(6,10)
T_{OF}	Output Float for all outputs		3	22	ns	(6)
Synchronous Inputs (1,9,10)						
T_{IS}	Input Setup					
	TIS1	D31:0	3		ns	
	TIS2	$\overline{BOFF}$	17		ns	
	TIS3	$\overline{BTERM}$/READY	7		ns	
	TIS4	HOLD	7		ns	
T_{IH}	Input Hold					
	TIH1	D31:0	5		ns	
	TIH2	$\overline{BOFF}$	5		ns	
	TIH3	$\overline{BTERM}$/READY	2		ns	
	TIH4	HOLD	3		ns	
Relative Output Timings (1,2,3,8)						
T_{AVSH1}	A31:2 Valid to $\overline{ADS}$ Rising		T – 4	T + 4	ns	
T_{AVSH2}	$\overline{BE3:0}$, W/$\overline{R}$, $\overline{SUP}$, D/$\overline{C}$, $\overline{DMA}$, $\overline{DACK3:0}$ Valid to $\overline{ADS}$ Rising		T – 6	T + 6	ns	
T_{AVEL1}	A31:2 Valid to $\overline{DEN}$ Falling		T – 4	T + 4	ns	
T_{AVEL2}	$\overline{BE3:0}$, W/$\overline{R}$, $\overline{SUP}$, $\overline{INST}$,$\overline{DMA}$, $\overline{DACK3:0}$ Valid to $\overline{DEN}$ Falling		T – 6	T + 6	ns	
T_{NLQV}	$\overline{WAIT}$ Falling to Output Data Valid		± 6		ns	

NOTES:
See Table 18 for all notes related to AC specifications.

Table 16. 80960CF AC Characteristics (33 MHz) (Sheet 3 of 3)

(80960CF-33 only, per the conditions in **4.2 Operating Conditions** and **4.5.1 AC TEST CONDITIONS.**)

Symbol	Parameter	Min	Max	Units	Notes
T_{DVNH}	Output Data Valid to $\overline{WAIT}$ Rising	N*T – 6	N*T + 6	ns	(4)
T_{NLNH}	$\overline{WAIT}$ Falling to $\overline{WAIT}$ Rising	N*T ± 4		ns	(4)
T_{NHQX}	Output Data Hold after $\overline{WAIT}$ Rising	(N+1)*T–8	(N+1)*T+6	ns	(5)
T_{EHTV}	DT/$\overline{R}$ Hold after $\overline{DEN}$ High	T/2 – 7	∞	ns	(6)
T_{TVEL}	DT/$\overline{R}$ Valid to $\overline{DEN}$ Falling	T/2 – 4		ns	
Relative Input Timings (1,2,3)					
T_{IS5}	$\overline{RESET}$ Input Setup (2-x Clock Mode)	6		ns	(13)
T_{IH5}	$\overline{RESET}$ Input Hold (2-x Clock Mode)	5		ns	(13)
T_{IS6}	$\overline{DREQ3:0}$ Input Setup	12		ns	(7)
T_{IH6}	$\overline{DREQ3:0}$ Input Hold	7		ns	(7)
T_{IS7}	$\overline{XINT7:0}$, NMI Input Setup	7		ns	(15)
T_{IH7}	$\overline{XINT7:0}$, NMI Input Hold	3		ns	(15)
T_{IS8}	$\overline{RESET}$ Input Setup (1-x Clock Mode)	3		ns	(14)
T_{IH8}	$\overline{RESET}$ Input Hold (1-x Clock Mode)	T/4 + 1		ns	(14)

NOTES:
See Table 18 for all notes related to AC specifications.

		In 1-x Mode (f_{CLK1x})	0	62.5	ns	(11)
		In 2-x Mode (f_{CLK2x})	8	∞	ns	
T_{CL}	CLKIN Low Time	In 1-x Mode (f_{CLK1x})	8	62.5	ns	(11)
		In 2-x Mode (f_{CLK2x})	8	∞	ns	
T_{CR}	CLKIN Rise Time		0	6	ns	
T_{CF}	CLKIN Fall Time		0	6	ns	
Output Clocks (1,8)						
T_{CP}	CLKIN to PCLK2:1 Delay	In 1-x Mode (f_{CLK1x})	– 2	2	ns	(3,12)
		In 2-x Mode (f_{CLK2x})	2	25	ns	(3)
T	PCLK2:1 Period	In 1-x Mode (f_{CLK1x})	T_C		ns	(12)
		In 2-x Mode (f_{CLK2x})	$2T_C$		ns	(3)
T_{PH}	PCLK2:1 High Time		(T/2) – 3	T/2	ns	(12)
T_{PL}	PCLK2:1 Low Time		(T/2) – 3	T/2	ns	(12)
T_{PR}	PCLK2:1 Rise Time		1	4	ns	(3)
T_{PF}	PCLK2:1 Fall Time		1	4	ns	(3)
Synchronous Outputs (8)						
T_{OH}	Output Valid Delay, Output Hold					(6,10)
T_{OV}	T_{OH1}, T_{OV1}	A31:2	3	16	ns	
	T_{OH2}, T_{OV2}	$\overline{BE3:0}$	3	18	ns	
	T_{OH3}, T_{OV3}	$\overline{ADS}$	6	20	ns	
	T_{OH4}, T_{OV4}	W/$\overline{R}$	3	20	ns	
	T_{OH5}, T_{OV5}	D/$\overline{C}$, $\overline{SUP}$, $\overline{DMA}$	4	18	ns	
	T_{OH6}, T_{OV6}	$\overline{BLAST}$, $\overline{WAIT}$	5	18	ns	
	T_{OH7}, T_{OV7}	$\overline{DEN}$	3	18	ns	
	T_{OH8}, T_{OV8}	HOLDA, BREQ	4	18	ns	
	T_{OH9}, T_{OV9}	$\overline{LOCK}$	4	18	ns	
	T_{OH10}, T_{OV10}	$\overline{DACK3:0}$	4	20	ns	
	T_{OH11}, T_{OV11}	D31:0	3	18	ns	
	T_{OH12}, T_{OV12}	DT/$\overline{R}$	T/2 + 3	T/2 + 16	ns	
	T_{OH13}, T_{OV13}	$\overline{FAIL}$	2	16	ns	
	T_{OH14}, T_{OV14}	$\overline{EOP3:0}$/$\overline{TC3:0}$	3	20	ns	(6,10)
T_{OF}	Output Float for all outputs		3	22	ns	(6)

NOTES: See Table 18 for all notes related to AC specifications.

Table 17. 80960CF AC Characteristics (25 MHz) (Sheet 2 of 2)

(80960CF-25 only, per the conditions in **4.2 Operating Conditions** and **4.5.1 AC TEST CONDITIONS**.)

Symbol	Parameter	Min	Max	Unit	Notes
Synchronous Inputs (1,9,10)					
T_{IS}	Input Setup				
T_{IS1}	D31:0	5		ns	
T_{IS2}	$\overline{BOFF}$	19		ns	
T_{IS3}	$\overline{BTERM}$/READY	9		ns	
T_{IS4}	HOLD	9		ns	
T_{IH}	Input Hold				
T_{IH1}	D31:0	5		ns	
T_{IH2}	$\overline{BOFF}$	7		ns	
T_{IH3}	$\overline{BTERM}$/READY	2		ns	
T_{IH4}	HOLD	5		ns	
Relative Output Timings (1,2,3,8)					
T_{AVSH1}	A31:2 Valid to $\overline{ADS}$ Rising	T – 4	T + 4	ns	
T_{AVSH2}	$\overline{BE3:0}$, W/$\overline{R}$, $\overline{SUP}$, D/$\overline{C}$, $\overline{DMA}$, $\overline{DACK3:0}$ Valid to $\overline{ADS}$ Rising	T – 6	T + 6	ns	
T_{AVEL1}	A31:2 Valid to $\overline{DEN}$ Falling	T – 4	T + 4	ns	
T_{AVEL2}	$\overline{BE3:0}$, W/$\overline{R}$, $\overline{SUP}$, $\overline{INST}$, $\overline{DMA}$, $\overline{DACK3:0}$ Valid to $\overline{DEN}$ Falling	T – 6	T + 6	ns	
T_{NLQV}	$\overline{WAIT}$ Falling to Output Data Valid	± 6		ns	
T_{DVNH}	Output Data Valid to $\overline{WAIT}$ Rising	N*T – F	N*T + 6	ns	(4)
T_{NLNH}	$\overline{WAIT}$ Falling to $\overline{WAIT}$ Rising	N*T ± 4		ns	(4)
T_{NHQX}	Output Data Hold after $\overline{WAIT}$ Rising	(N+1)*T– 8	(N+1)*T+6	ns	(5)
T_{EHTV}	DT/$\overline{R}$ Hold after $\overline{DEN}$ High	T/2 – 7	∞	ns	(6)
T_{TVEL}	DT/$\overline{R}$ Valid to $\overline{DEN}$ Falling	T/2 – 4		ns	
Relative Input Timings (1,2,3)					
T_{IS5}	$\overline{RESET}$ Input Setup (2-x Clock Mode)	8		ns	(13)
T_{IH5}	$\overline{RESET}$ Input Hold (2-x Clock Mode)	7		ns	(13)
T_{IS6}	$\overline{DREQ3:0}$ Input Setup	14		ns	(7)
T_{IH6}	$\overline{DREQ3:0}$ Input Hold	9		ns	(7)
T_{IS7}	$\overline{XINT7:0}$, $\overline{NMI}$ Input Setup	9		ns	(15)
T_{IH7}	$\overline{XINT7:0}$, $\overline{NMI}$ Input Hold	5		ns	(15)
T_{IS8}	$\overline{RESET}$ Input Setup (1-x Clock Mode)	3		ns	(14)
T_{IH8}	$\overline{RESET}$ Input Hold (1-x Clock Mode)	T/4 + 1		ns	(14)

NOTES: See Table 18 for all notes related to AC specifications.

Table 18. 80960CF AC Characteristics (16 MHz) (Sheet 1 of 3)

(80960CF-16 only, per the conditions in **4.2 Operating Conditions** and **4.5.1 AC TEST CONDITIONS**.)

Symbol	Parameter		Min	Max	Units	Notes
Input Clock (1,9)						
T_F	CLKIN Frequency		0	32	MHz	
T_C	CLKIN Period	In 1-x Mode (f_{CLK1x}) In 2-x Mode (f_{CLK2x})	62.5 31.25	125 ∞	ns ns	(11)
T_{CS}	CLKIN Period Stability	In 1-x Mode (f_{CLK1x})		±0.1%	Δ	(12)
T_{CH}	CLKIN High Time	In 1-x Mode (f_{CLK1x}) In 2-x Mode (f_{CLK2x})	10 10	62.5 ∞	ns ns	(11)
T_{CL}	CLKIN Low Time	In 1-x Mode (f_{CLK1x}) In 2-x Mode (f_{CLK2x})	10 10	62.5 ∞	ns ns	(11)
T_{CR}	CLKIN Rise Time		0	6	ns	
T_{CF}	CLKIN Fall Time		0	6	ns	
Output Clocks (1,8)						
T_{CP}	CLKIN to PCLK2:1 Delay	In 1-x Mode (f_{CLK1x}) In 2-x Mode (f_{CLK2x})	−2 2	2 25	ns ns	(3,12) (3)
T	PCLK2:1 Period	In 1-x Mode (f_{CLK1x}) In 2-x Mode (f_{CLK2x})	TC 2TC		ns ns	(12) (3)
T_{PH}	PCLK2:1 High Time		(T/2) − 4	T/2	ns	(12)
T_{PL}	PCLK2:1 Low Time		(T/2) − 4	T/2	ns	(12)
T_{PR}	PCLK2:1 Rise Time		1	4	ns	(3)
T_{PF}	PCLK2:1 Fall Time		1	4	ns	(3)
Synchronous Outputs (8)						
T_{OH} T_{OV}	Output Valid Delay, Output Hold					(6,10)
	T_{OH1}, T_{OV1}	A31:2	3	18	ns	
	T_{OH2}, T_{OV2}	$\overline{BE3:0}$	3	20	ns	
	T_{OH3}, T_{OV3}	$\overline{ADS}$	6	22	ns	
	T_{OH4}, T_{OV4}	W/$\overline{R}$	3	22	ns	
	T_{OH5}, T_{OV5}	D/$\overline{C}$, $\overline{SUP}$, $\overline{DMA}$	4	20	ns	
	T_{OH6}, T_{OV6}	$\overline{BLAST}$, $\overline{WAIT}$	5	20	ns	
	T_{OH7}, T_{OV7}	$\overline{DEN}$	3	20	ns	
	T_{OH8}, T_{OV8}	HOLDA, BREQ	4	20	ns	
	T_{OH9}, T_{OV9}	$\overline{LOCK}$	4	20	ns	
	T_{OH10}, T_{OV10}	$\overline{DACK3:0}$	4	22	ns	
	T_{OH11}, T_{OV11}	D31:0	3	20	ns	
	T_{OH12}, T_{OV12}	DT/$\overline{R}$	T/2 + 3	T/2 + 18	ns	
	T_{OH13}, T_{OV13}	$\overline{FAIL}$	2	18	ns	
	T_{OH14}, T_{OV14}	$\overline{EOP3:0}$/TC3:0	3	22	ns	(6,10)
T_{OF}	Output Float for all outputs		3	22	ns	(6)

Table 18. 80960CF AC Characteristics (16 MHz) (Sheet 2 of 3)

(80960CF-16 only, per the conditions in **4.2 Operating Conditions** and **4.5.1 AC TEST CONDITIONS.**)

Symbol	Parameter		Min	Max	Units	Notes
Synchronous Inputs (1,9,10)						
T_{IS}	Input Setup					
T_{IS1}		D31:0	5		ns	
T_{IS2}		$\overline{BOFF}$	21		ns	
T_{IS3}		$\overline{BTERM}$/READY	9		ns	
T_{IS4}		HOLD	9		ns	
T_{IH}	Input Hold					
T_{IH1}		D31:0	5		ns	
T_{IH2}		$\overline{BOFF}$	7		ns	
T_{IH3}		$\overline{BTERM}$/READY	2		ns	
T_{IH4}		HOLD	5		ns	
Relative Output Timings (1,2,3,8)						
T_{AVSH1}	A31:2 Valid to $\overline{ADS}$ Rising		T – 4	T + 4	ns	
T_{AVSH2}	$\overline{BE3:0}$, W/$\overline{R}$, $\overline{SUP}$, D/$\overline{C}$, DMA, $\overline{DACK3:0}$ Valid to $\overline{ADS}$ Rising		T – 6	T + 6	ns	
T_{AVEL1}	A31:2 Valid to $\overline{DEN}$ Falling		T – 6	T + 6	ns	
T_{AVEL2}	$\overline{BE3:0}$, W/$\overline{R}$, $\overline{SUP}$, $\overline{INST}$, DMA, $\overline{DACK3:0}$ Valid to $\overline{DEN}$ Falling		T – 6	T + 6	ns	
T_{NLQV}	$\overline{WAIT}$ Falling to Output Data Valid		± 6		ns	
T_{DVNH}	Output Data Valid to $\overline{WAIT}$ Rising		N*T – 6	N*T + 6	ns	(4)
T_{NLNH}	$\overline{WAIT}$ Falling to $\overline{WAIT}$ Rising		N*T ± 4		ns	(4)
T_{NHQX}	Output Data Hold after $\overline{WAIT}$ Rising		(N+1)*T–8	(N+1)*T+6	ns	(5)
T_{EHTV}	DT/$\overline{R}$ Hold after $\overline{DEN}$ High		T/2 – 7	∞	ns	(6)
T_{TVEL}	DT/$\overline{R}$ Valid to $\overline{DEN}$ Falling		T/2 – 4		ns	
Relative Input Timings (1,2,3)						
T_{IS5}	$\overline{RESET}$ Input Setup (2-x Clock Mode)		10		ns	(13)
T_{IH5}	$\overline{RESET}$ Input Hold (2-x Clock Mode)		9		ns	(13)
T_{IS6}	$\overline{DREQ3:0}$ Input Setup		16		ns	(7)
T_{IH6}	$\overline{DREQ3:0}$ Input Hold		11		ns	(7)
T_{IS7}	$\overline{XINT7:0}$, NMI Input Setup		9		ns	(15)

NOTES:

1. See **Section 4.5.2, AC TIMING WAVEFORMS** for waveforms and definitions.
2. See Figure 16 for capacitive derating information for output delays and hold times.
3. See Figure 17 for capacitive derating information for rise and fall times.
4. Where N is the number of NRAD, NRDD, NWAD or NWDD wait states that are programmed in the Bus Controller Region Table. WAIT never goes active when there are no wait states in an access.
5. N = Number of wait states inserted with READY.
6. Output Data and/or DT/R may be driven indefinitely following a cycle if there is no subsequent bus activity.
7. Since asynchronous inputs are synchronized internally by the 80960CF, they have no required setup or hold times to be recognized and for proper operation. However, to guarantee recognition of the input at a particular edge of PCLK2:1, the setup times shown must be met. Asynchronous inputs must be active for at least two consecutive PCLK2:1 rising edges to be seen by the processor.
8. These specifications are guaranteed by the processor.
9. These specifications must be met by the system for proper operation of the processor.
10. This timing is dependent upon the loading of PCLK2:1. Use the derating curves of **Section 4.5.3, DERATING CURVES** to adjust the timing for PCLK2:1 loading.
11. In the 1-x input clock mode, the maximum input clock period is limited to 125 ns while the processor is operating. When the processor is in reset, the input clock may stop even in 1-x mode.
12. When in the 1-x input clock mode, these specifications assume a stable input clock with a period variation of less than ± 0.1% between adjacent cycles.
13. In 2-x clock mode, RESET is an asynchronous input which has no required setup and hold time for proper operation. However, to guarantee the device exits reset synchronized to a particular clock edge, the RESET pin must meet setup and hold times to the falling edge of the CLKIN. (See Figure 23.)
14. In 1-x clock mode, RESET is an asynchronous input which has no required setup and hold time for proper operation. However, to guarantee the device exits reset synchronized to a particular clock edge, the RESET pin must meet setup and hold times to the rising edge of the CLKIN. (See Figure 24.)
15. The interrupt pins are synchronized internally by the 80960CF. They have no required setup or hold times for proper operation. These pins are sampled by the interrupt controller every other clock and must be active for at least three consecutive PCLK2:1 rising edges when asserting them asynchronously. To guarantee recognition at a particular clock edge, the setup and hold times shown must be met for two consecutive PCLK2:1 rising edges.

4.5.1 AC TEST CONDITIONS

The AC Specifications in Section 4.5 are tested with the 50 pF load shown in Figure 7. Figure 16 shows how timings vary with load capacitance.

Specifications are measured at the 1.5V crossing point, unless otherwise indicated. Input waveforms are assumed to have a rise and fall time of ≤ 2 ns from 0.8V to 2.0V. See **Section 4.5.2, AC TIMING WAVEFORMS** for AC specification definitions, test points and illustrations.

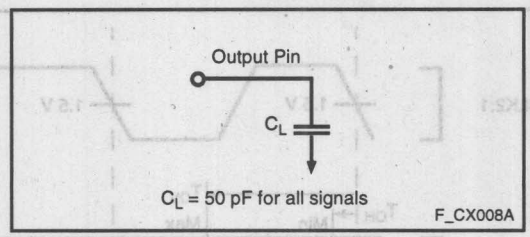


Figure 7. AC Test Load

4.5.2 AC TIMING WAVEFORMS

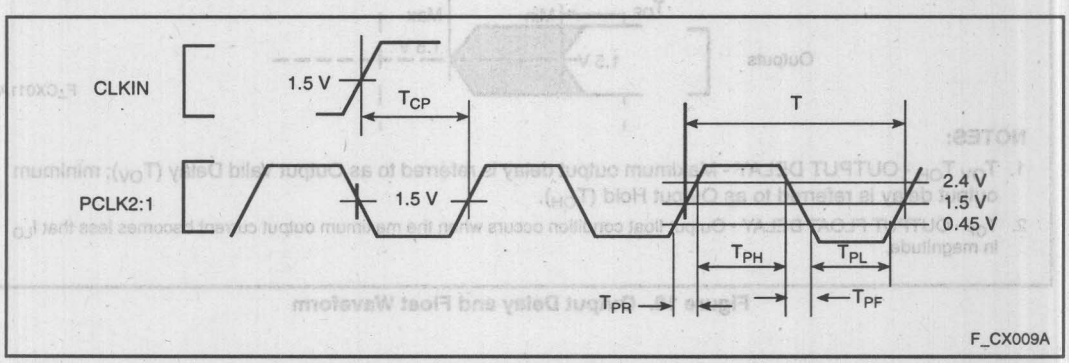


Figure 8. Input and Output Clocks Waveform

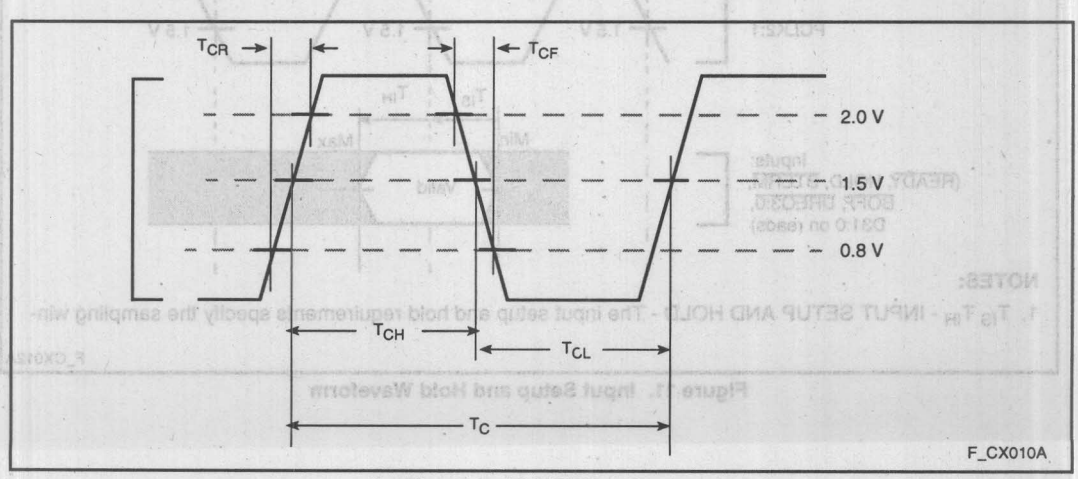
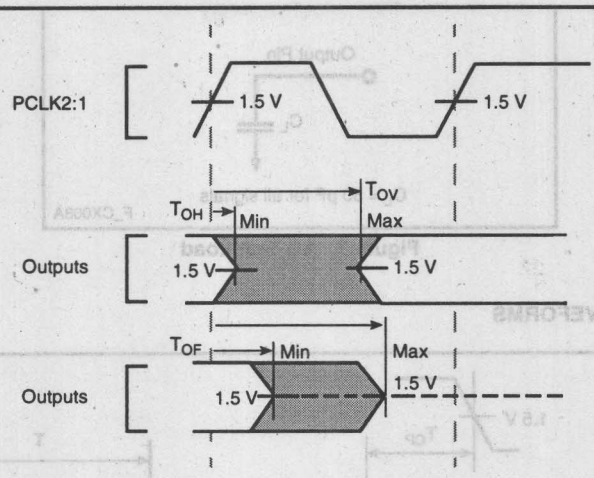


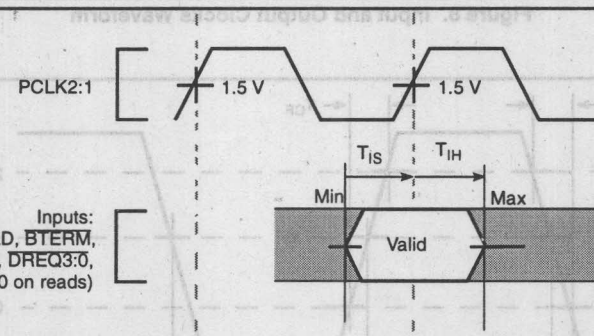
Figure 9. CLKIN Waveform



NOTES:

1. T_{OV} T_{OH} - OUTPUT DELAY - Maximum output delay is referred to as Output Valid Delay (T_{OV}); minimum output delay is referred to as Output Hold (T_{OH}).
2. T_{OF} - OUTPUT FLOAT DELAY - Output float condition occurs when the maximum output current becomes less that I_{LO} in magnitude.

Figure 10. Output Delay and Float Waveform



NOTES:

1. T_{IS} T_{IH} - INPUT SETUP AND HOLD - The input setup and hold requirements specify the sampling win-

Figure 11. Input Setup and Hold Waveform

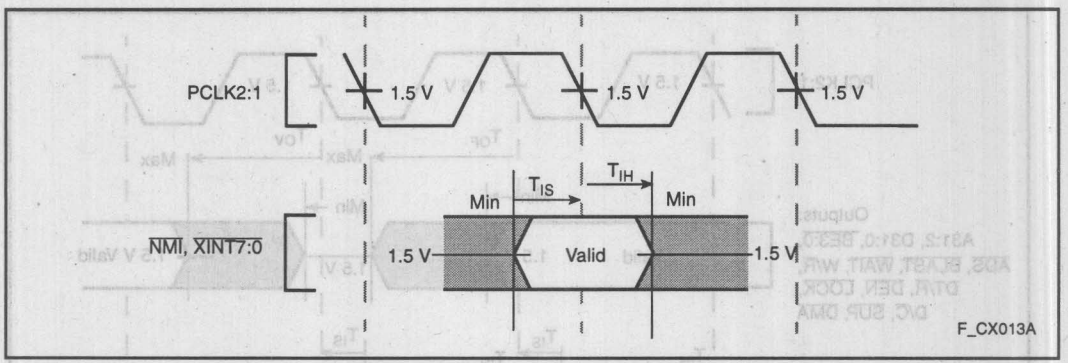
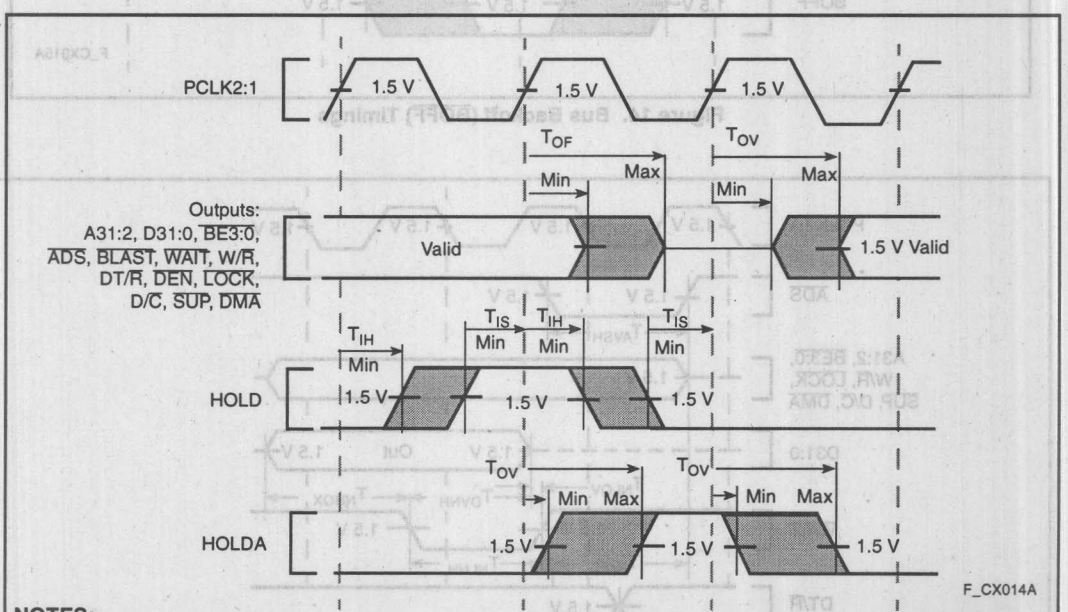


Figure 12. NMI, XINT7:0 Input Setup and Hold Waveform



NOTES:

1. T_{OV} T_{OH} - OUTPUT DELAY - Maximum output delay is referred to as Output Valid Delay (T_{OV}); minimum output delay is referred to as Output Hold (T_{OH}).
2. T_{OF} - OUTPUT FLOAT DELAY - Output float condition occurs when the maximum output current becomes less that I_{LO} in magnitude.
3. T_{IS} T_{IH} - INPUT SETUP AND HOLD - The input setup and hold requirements specify the sampling window during which synchronous inputs must be stable for correct processor operation.

Figure 13. Hold Acknowledge Timings

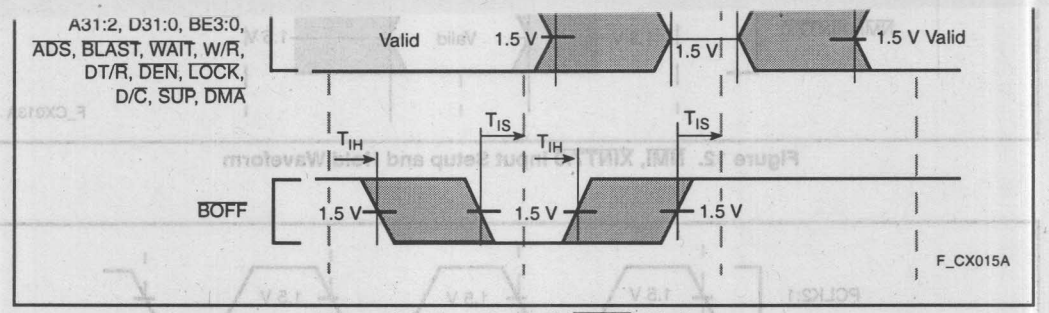


Figure 14. Bus Backoff ($\overline{BOFF}$) Timings

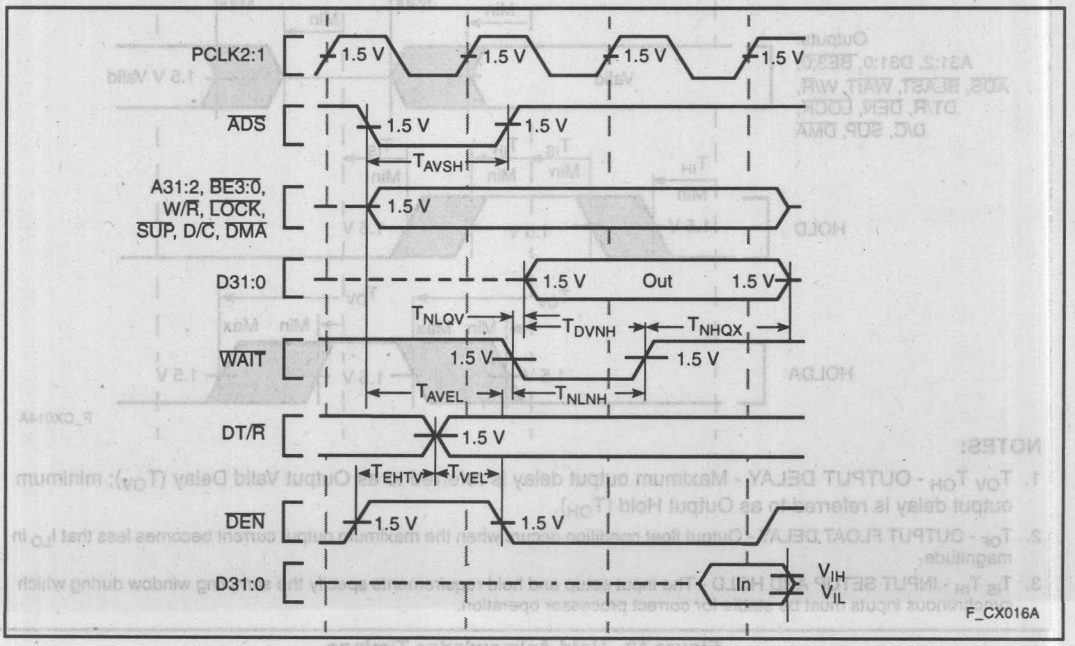


Figure 15. Relative Timings Waveforms

4.5.3 DERATING CURVES

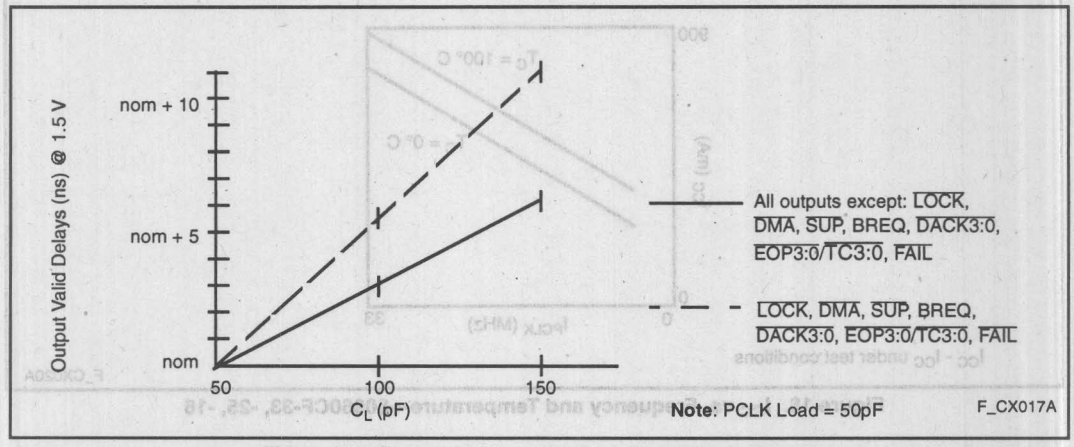


Figure 16. Output Delay or Hold vs. Load Capacitance

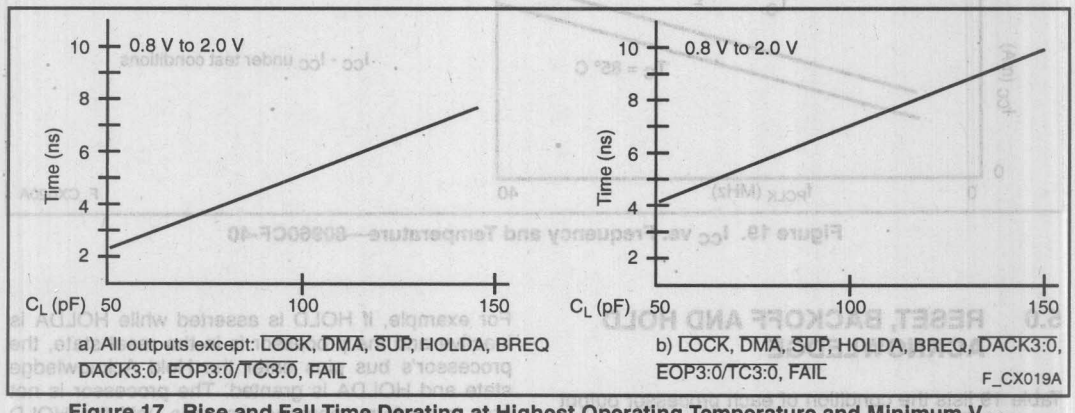


Figure 17. Rise and Fall Time Derating at Highest Operating Temperature and Minimum V_{CC}

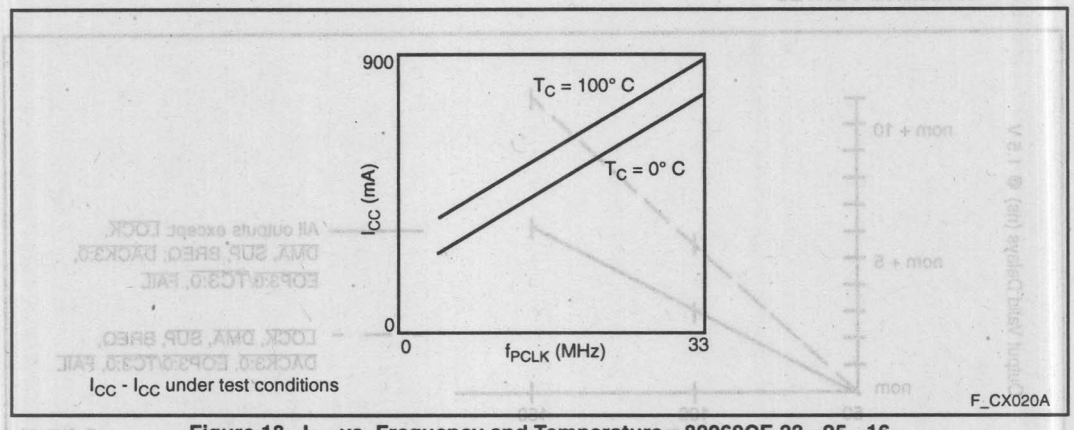


Figure 18. I_{CC} vs. Frequency and Temperature—80960CF-33, -25, -16

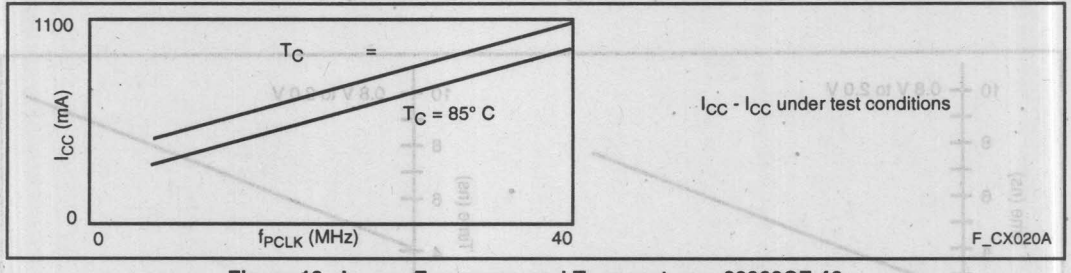


Figure 19. I_{CC} vs. Frequency and Temperature—80960CF-40

5.0 RESET, BACKOFF AND HOLD ACKNOWLEDGE

Table 19 lists the condition of each processor output pin while $\overline{\text{RESET}}$ is asserted (low). Table 20 lists the condition of each processor output pin while HOLDA is asserted (high).

In Table 20, with regard to bus output pin state only, the Hold Acknowledge state takes precedence over the reset state. Although asserting the $\overline{\text{RESET}}$ pin internally resets the processor, the processor's bus output pins do not enter the reset state if Hold Acknowledge has been granted to a previous HOLD request (HOLDA is active). Furthermore, the processor grants new HOLD requests and enters the Hold Acknowledge state even while in reset.

For example, if HOLD is asserted while HOLDA is inactive and the processor is in the reset state, the processor's bus pins enter the Hold Acknowledge state and HOLDA is granted. The processor is not able to perform memory accesses until the HOLD request is removed, even if the $\overline{\text{RESET}}$ pin is brought high. This operation is provided to simplify boot-up synchronization among multiple processors sharing the same bus.

Table 19. Reset Conditions

Pins	State During Reset (HOLDA inactive)
A31:2	Floating
D31:0	Floating
$\overline{BE3:0}$	Driven high (Inactive)
W/$\overline{R}$	Driven low (Read)
$\overline{ADS}$	Driven high (Inactive)
$\overline{WAIT}$	Driven high (Inactive)
$\overline{BLAST}$	Driven low (Active)
DT/$\overline{R}$	Driven low (Receive)
$\overline{DEN}$	Driven high (Inactive)
$\overline{LOCK}$	Driven high (Inactive)
BREQ	Driven low (Inactive)
D/$\overline{C}$	Floating
$\overline{DMA}$	Floating
$\overline{SUP}$	Floating
$\overline{FAIL}$	Driven low (Active)
$\overline{DACK3:0}$	Driven high (Inactive)
$\overline{EOP3:0}$/TC3:0	Floating (Set to input mode)

Table 20. Hold Acknowledge and Backoff Conditions

Pins	State During HOLDA
A31:2	Floating
D31:0	Floating
$\overline{BE3:0}$	Floating
W/$\overline{R}$	Floating
$\overline{ADS}$	Floating
$\overline{WAIT}$	Floating
$\overline{BLAST}$	Floating
DT/$\overline{R}$	Floating
$\overline{DEN}$	Floating
$\overline{LOCK}$	Floating
BREQ	Driven (High or low)
D/$\overline{C}$	Floating
$\overline{DMA}$	Floating
$\overline{SUP}$	Floating
$\overline{FAIL}$	Driven high (Inactive)
$\overline{DACK3:0}$	Driven high (Inactive)
$\overline{EOP3:0}$/TC3:0	Driven (If output)

6.0 BUS WAVEFORMS

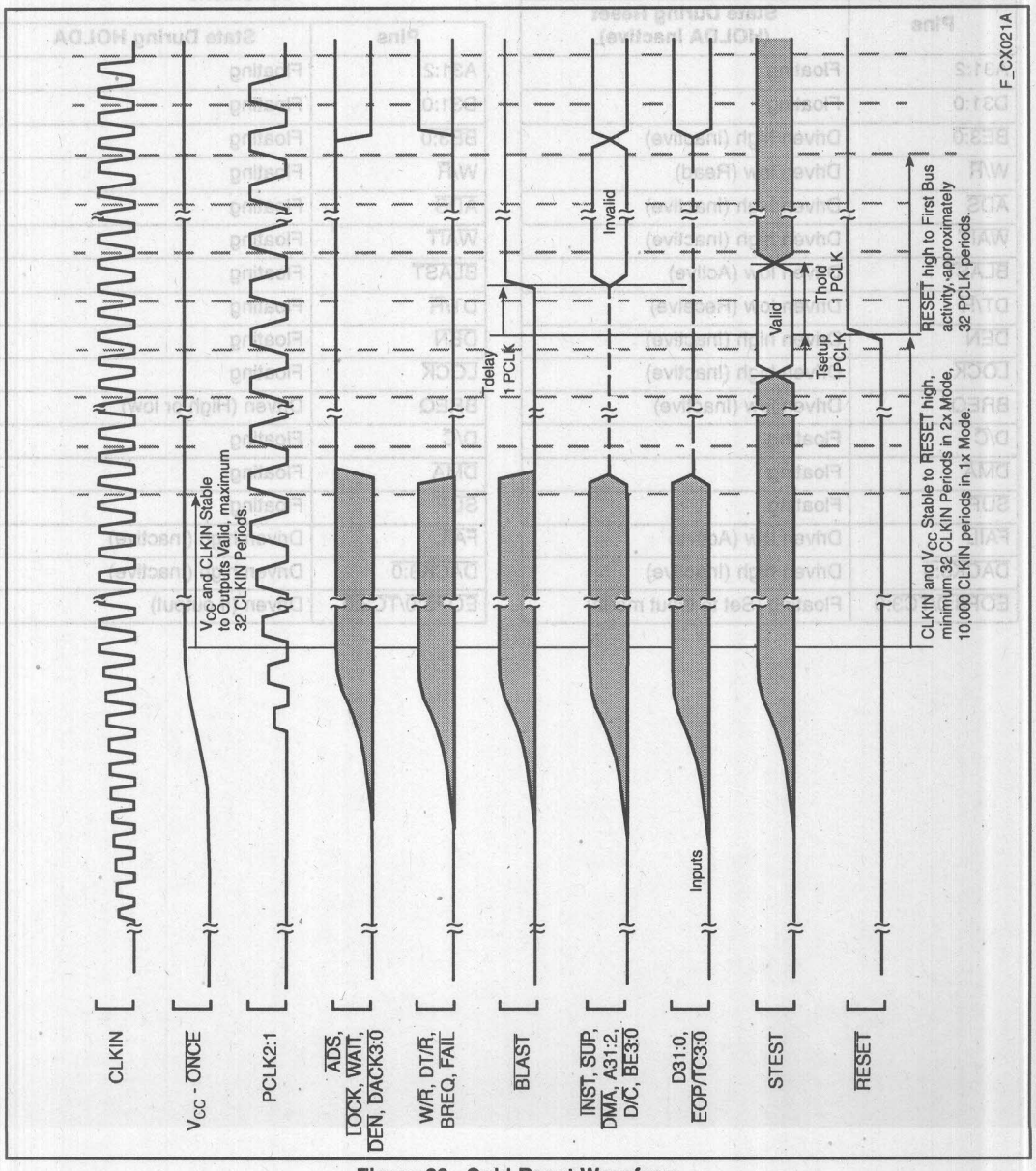


Figure 20. Cold Reset Waveform

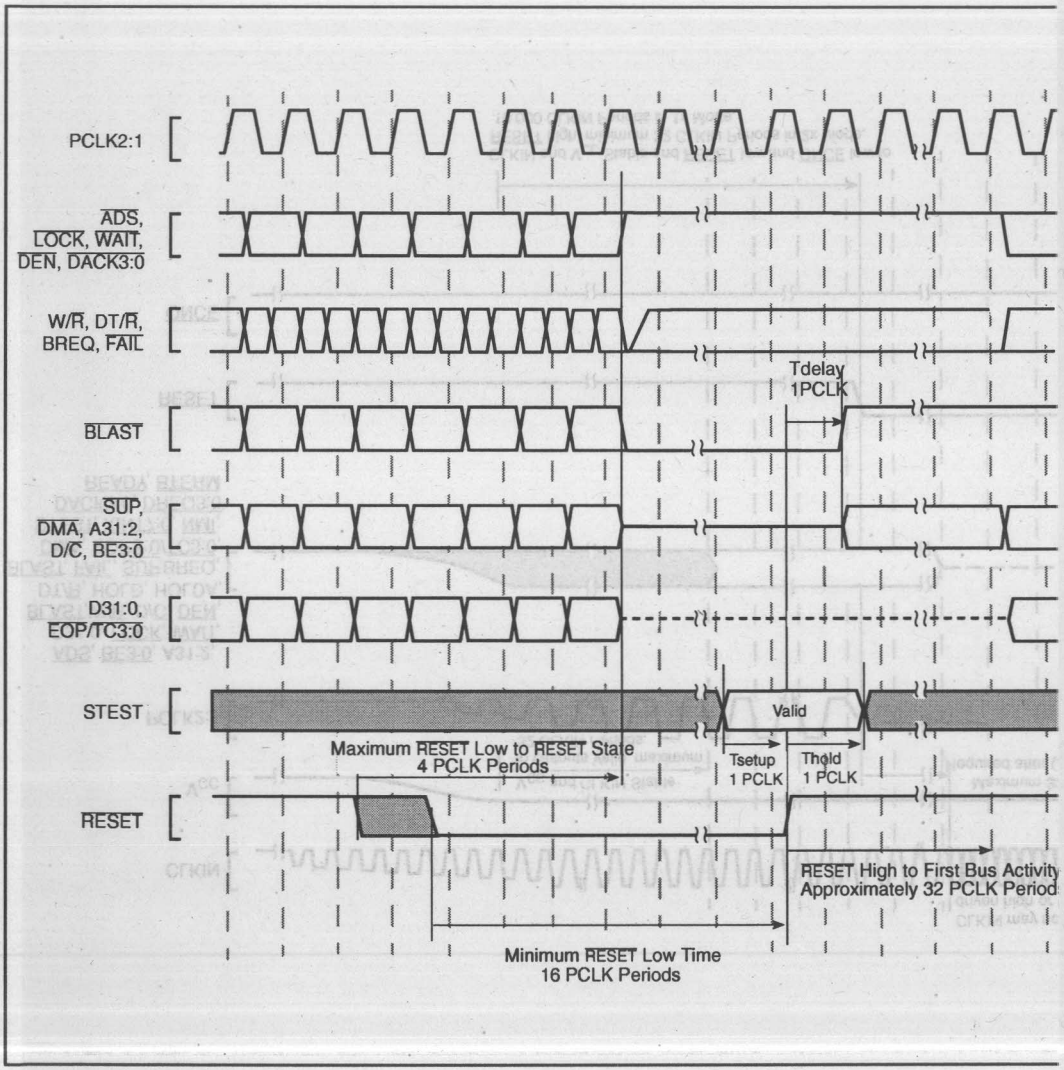


Figure 21. Warm Reset Waveform

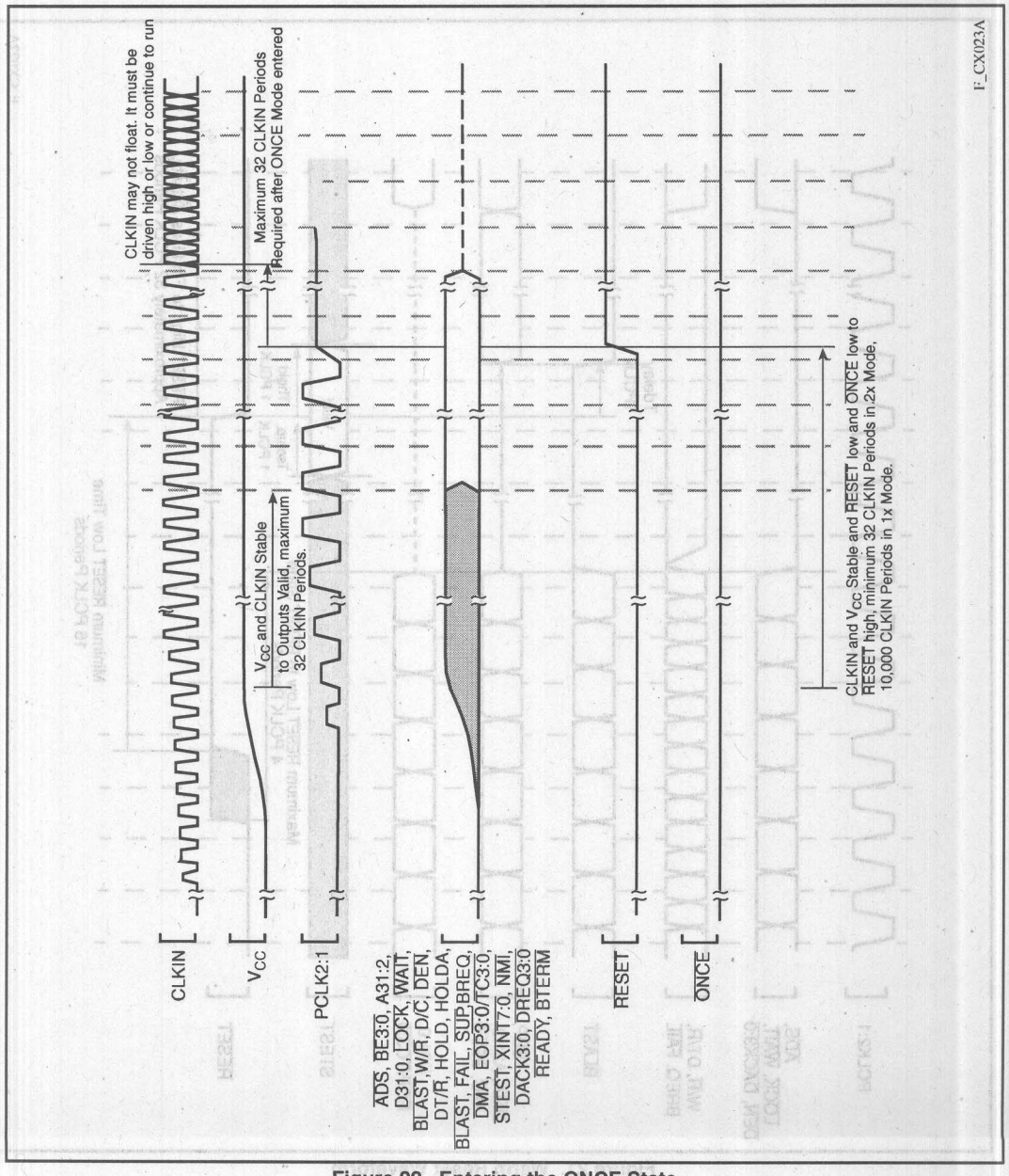
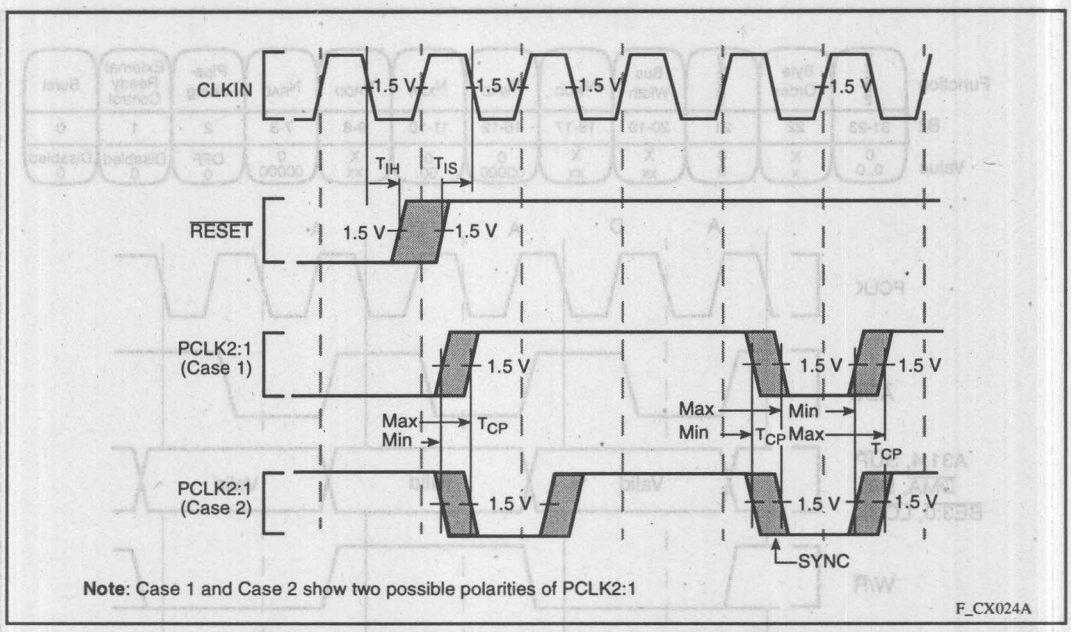


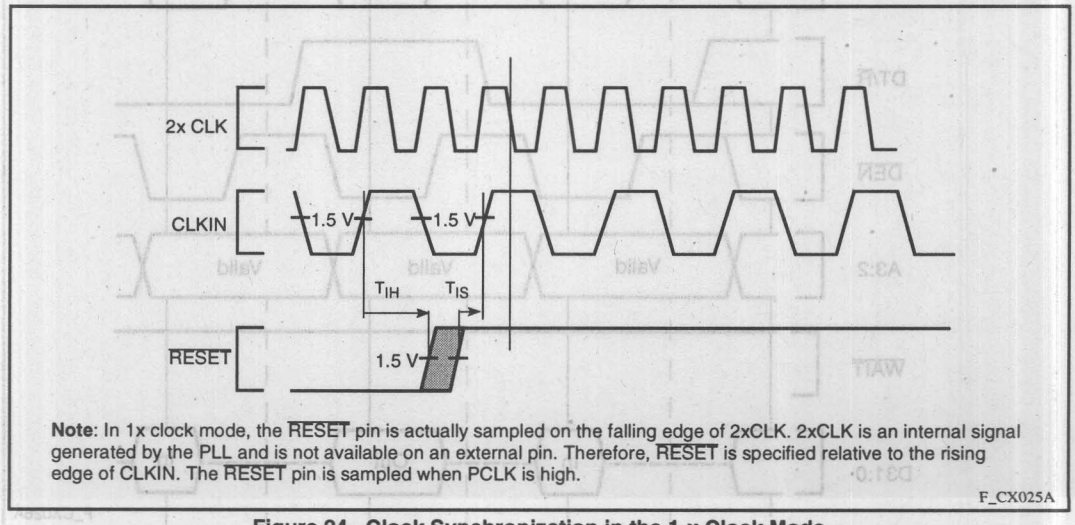
Figure 22. Entering the ONCE State



Note: Case 1 and Case 2 show two possible polarities of PCLK2:1

F_CX024A

Figure 23. Clock Synchronization in the 2-x Clock Mode



Note: In 1x clock mode, the RESET pin is actually sampled on the falling edge of 2xCLK. 2xCLK is an internal signal generated by the PLL and is not available on an external pin. Therefore, RESET is specified relative to the rising edge of CLKIN. The RESET pin is sampled when PCLK is high.

F_CX025A

Figure 24. Clock Synchronization in the 1-x Clock Mode

Function	reserved	Byte Order	reserved	Bus Width	N_{WDD}	N_{WAD}	N_{XDA}	N_{RDD}	N_{RAD}	Pipe-Lining	External Ready Control	Burst
Bit	31-23	22	21	20-19	18-17	16-12	11-10	9-8	7-3	2	1	0
Value	0 0..0	X x	0 0	X xx	X xx	0 00000	0 00	X xx	0 00000	OFF 0	Disabled 0	Disabled 0

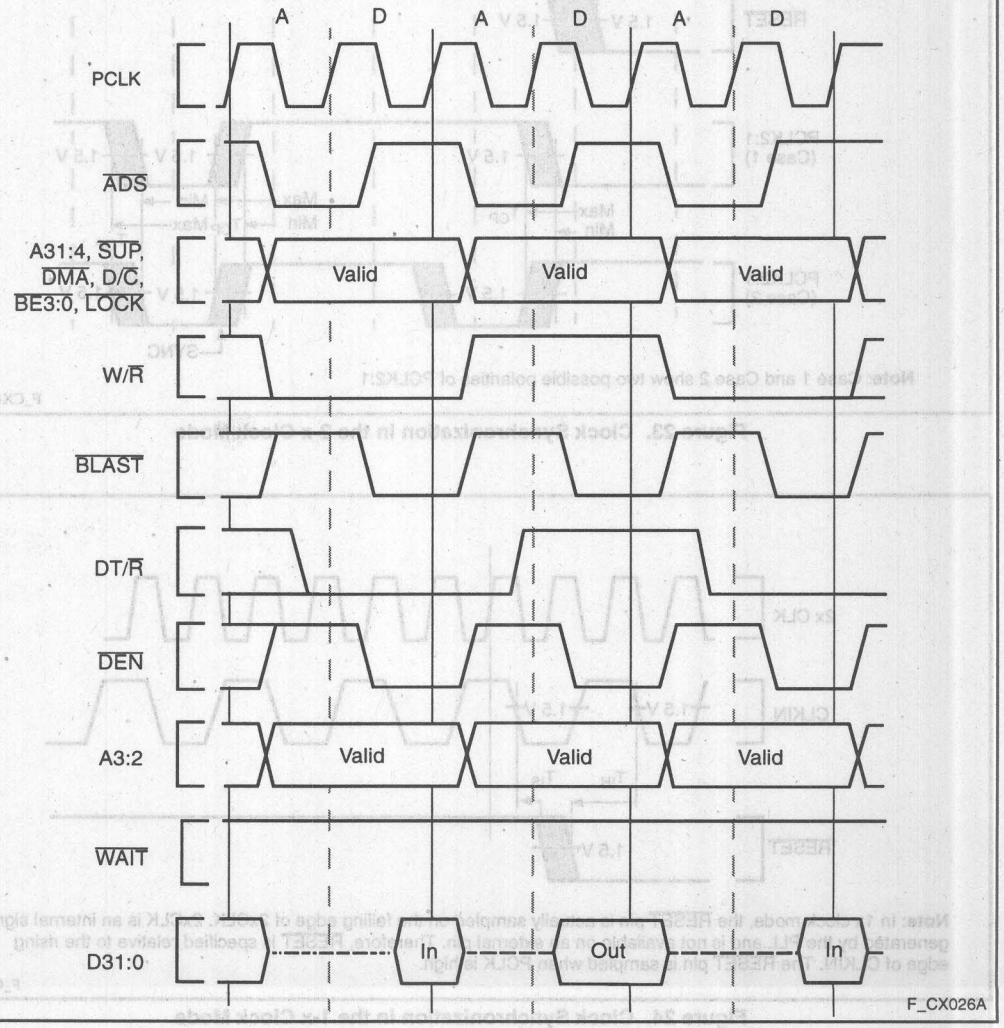


Figure 25. Non-Burst, Non-Pipelined Requests Without Wait States

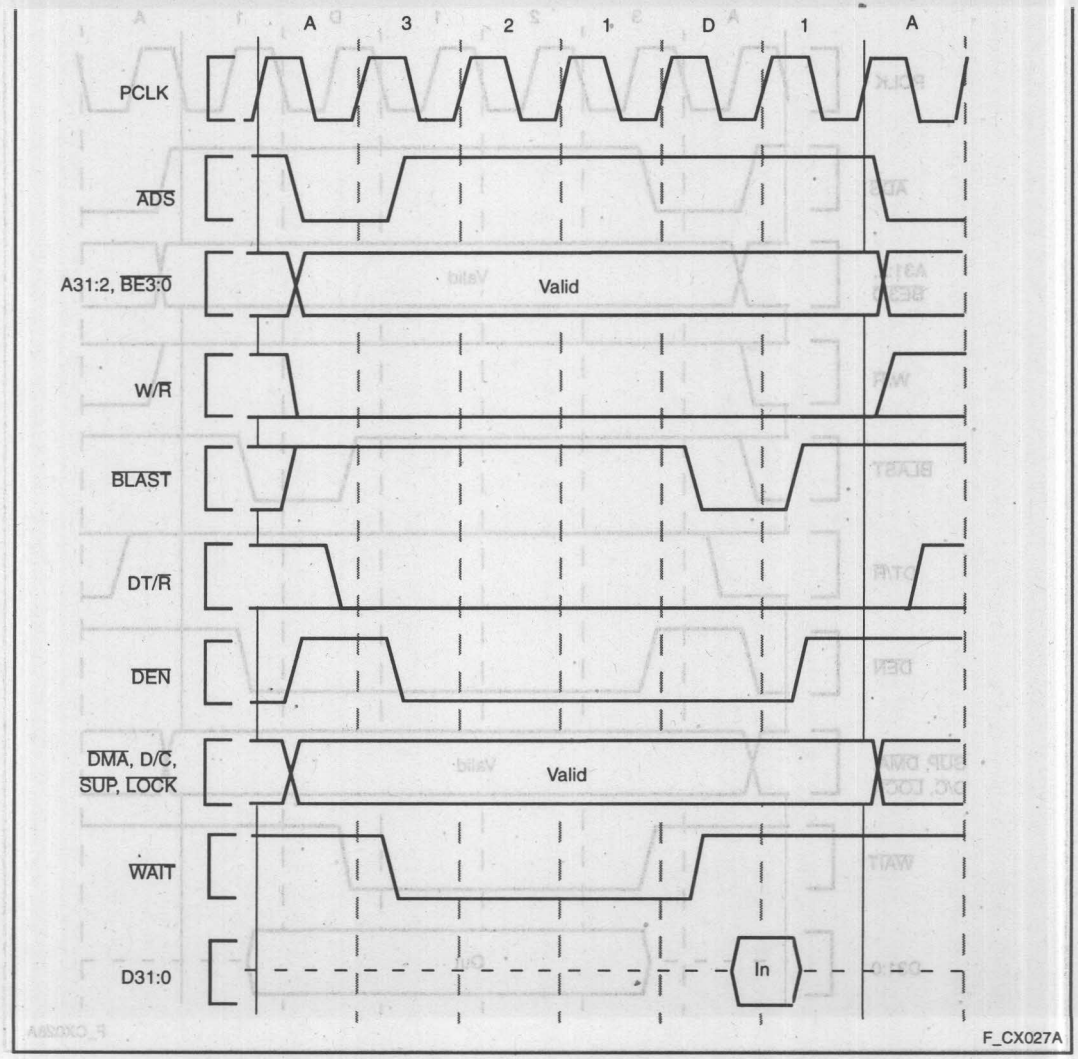


Figure 26. Non-Burst, Non-Pipelined Read Request With Wait States

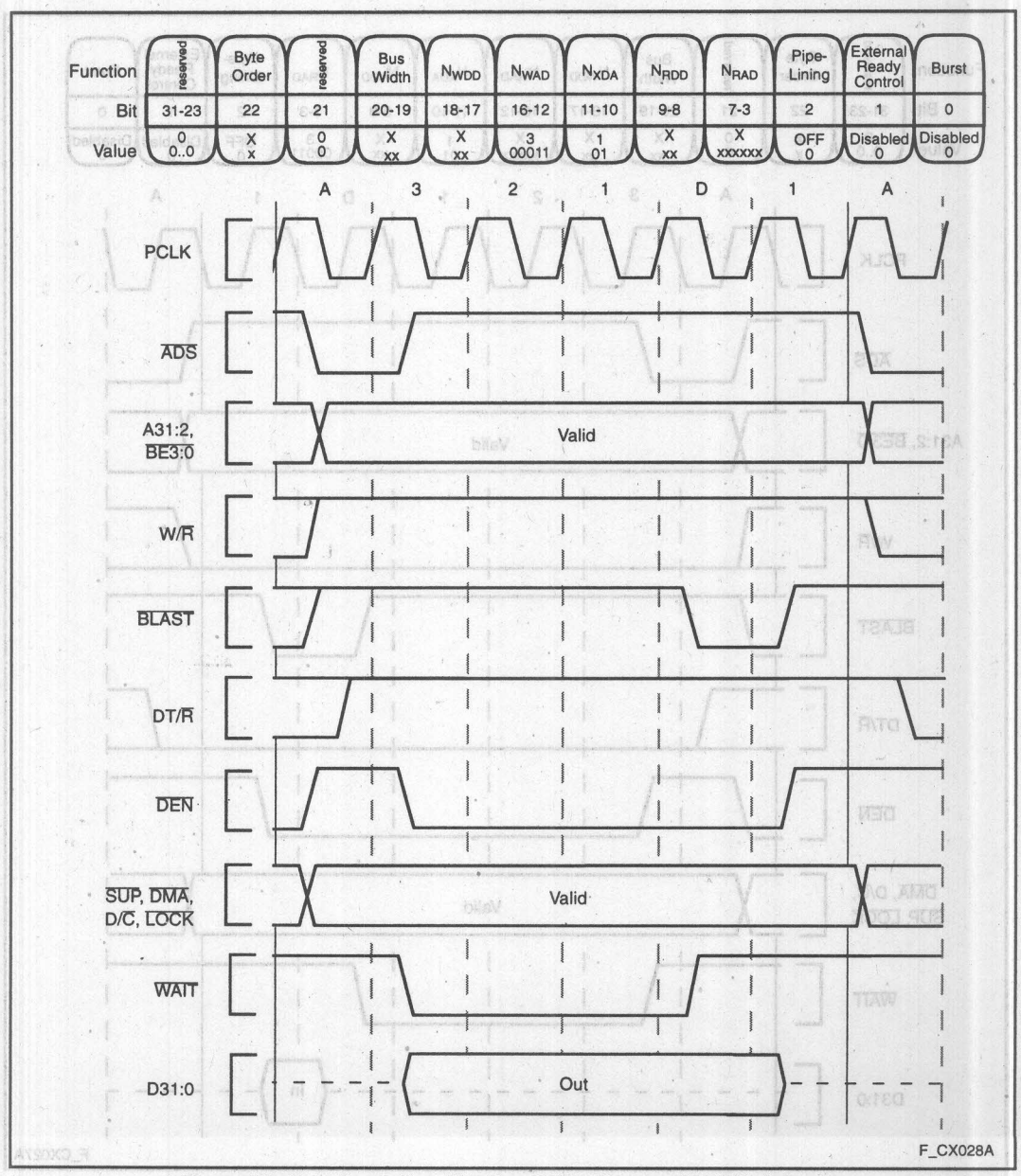


Figure 27. Non-Burst, Non-Pipelined Write Request With Wait States

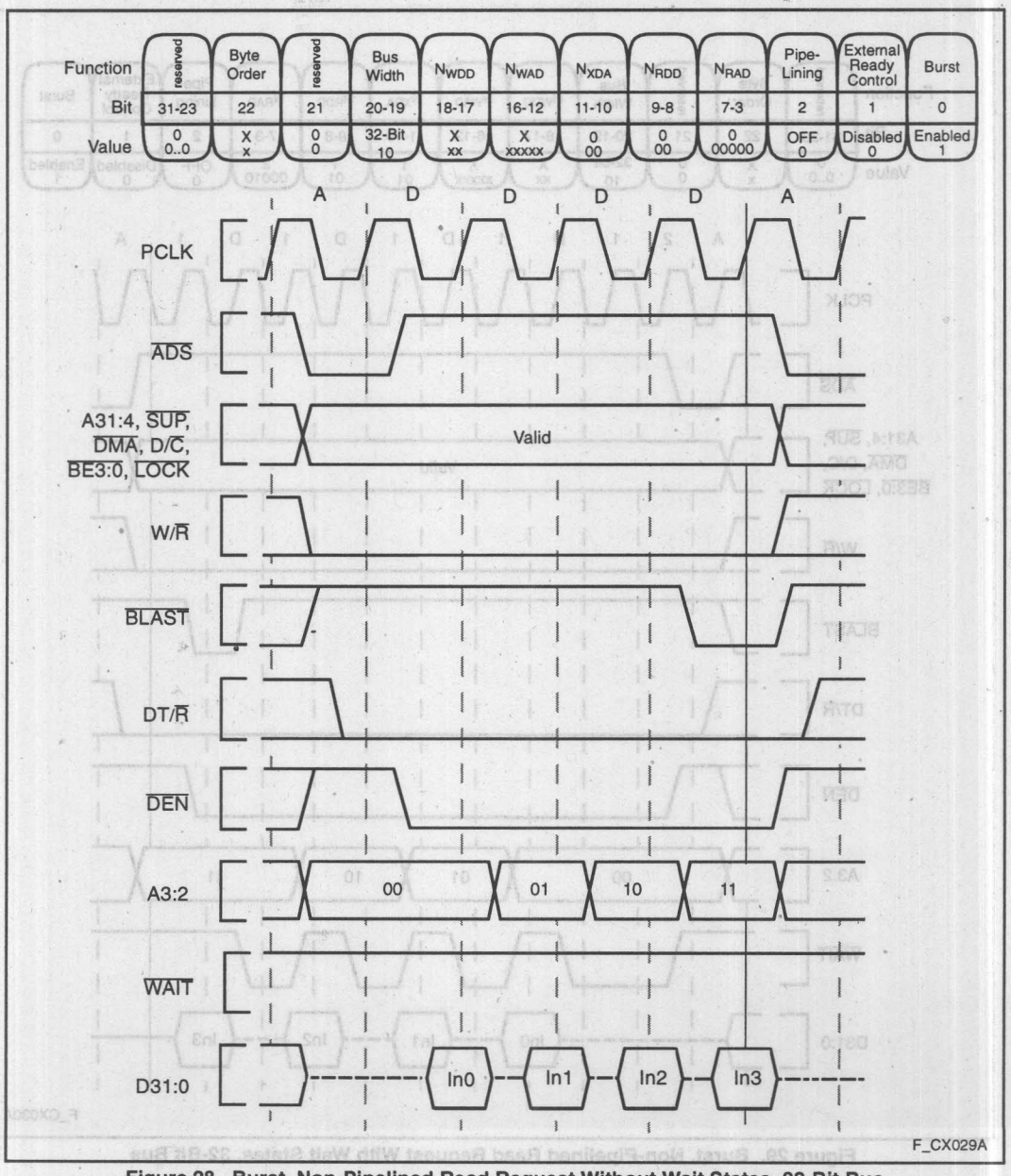


Figure 28. Burst, Non-Pipelined Read Request Without Wait States, 32-Bit Bus

Function	reserved	Byte Order	reserved	Bus Width	N_{WDD}	N_{WAD}	N_{XDA}	N_{RDD}	N_{RAD}	Pipe-Lining	External Ready Control	Burst
Bit	31-23	22	21	20-19	18-17	16-12	11-10	9-8	7-3	2	1	0
Value	0 0..0	X x	0 0	32-bit 10	X xx	X xxxxx	1 01	1 01	2 00010	OFF 0	Disabled 0	Enabled 1

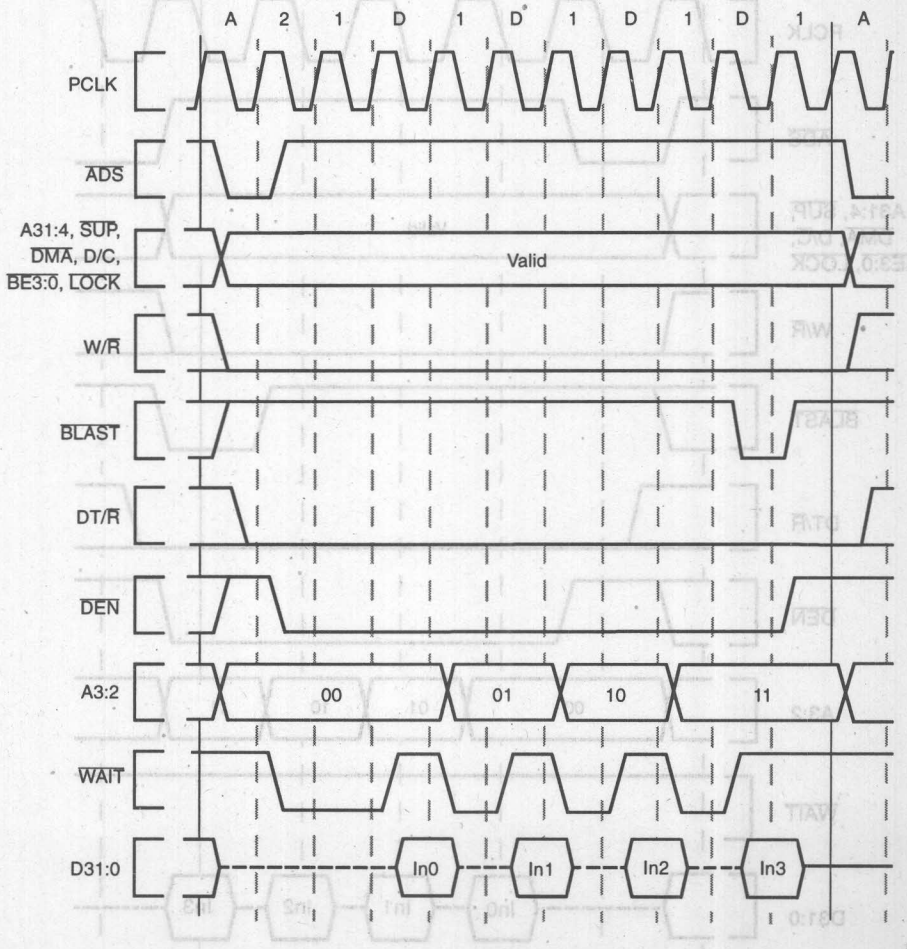


Figure 29. Burst, Non-Pipelined Read Request With Wait States, 32-Bit Bus

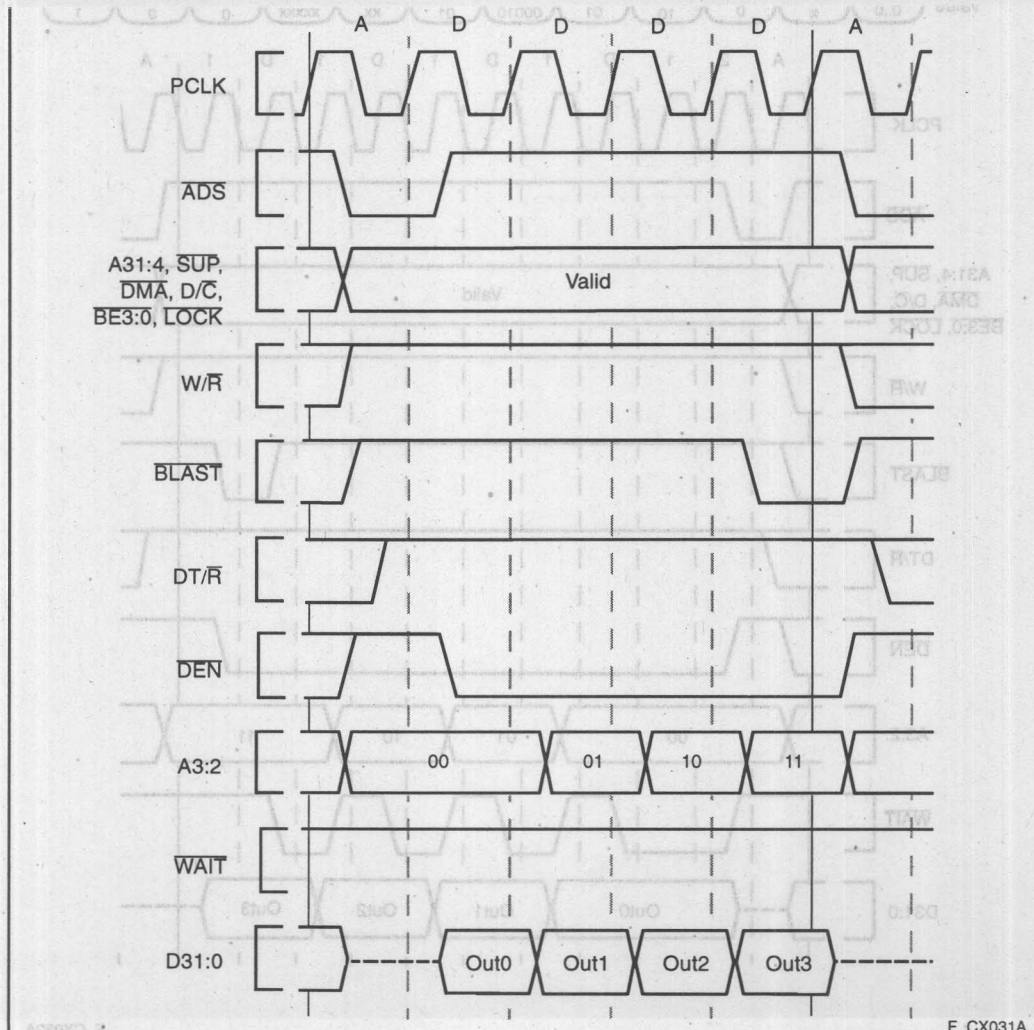


Figure 30. Burst, Non-Pipelined Write Request Without Wait States, 32-Bit Bus

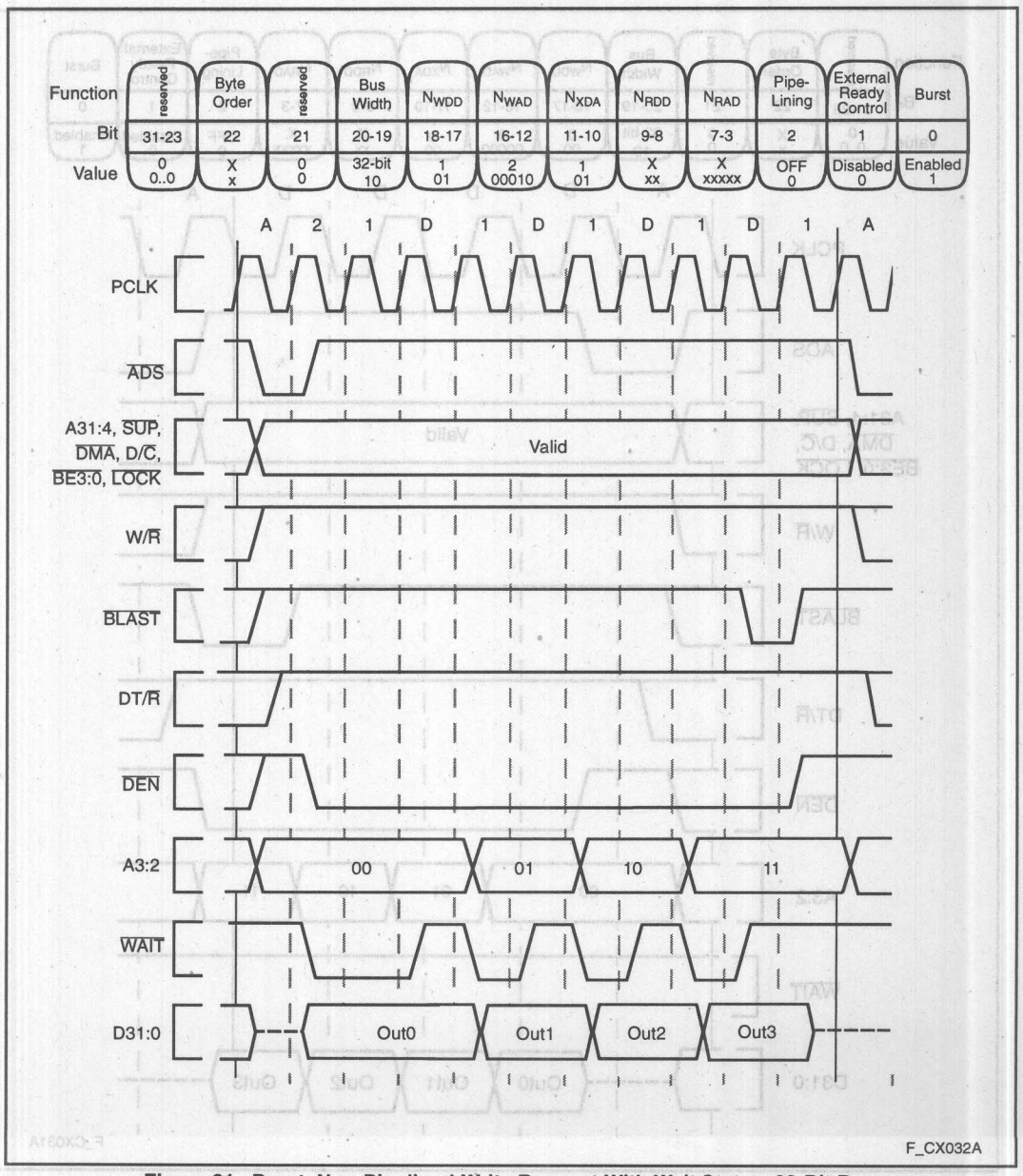


Figure 31. Burst, Non-Pipelined Write Request With Wait States, 32-Bit Bus

Function	reserved	Byte Order	reserved	Bus Width	N_{WDD}	N_{WAD}	N_{XDA}	N_{RDD}	N_{RAD}	Pipe-Lining	External Ready Control	Burst
Bit	31-23	22	21	20-19	18-17	16-12	11-10	9-8	7-3	2	1	0
Value	0 0..0	X x	0 0	16-bit 01	X xx	X xxxxx	1 01	1 01	2 00010	OFF 0	Disabled 0	Enabled 1

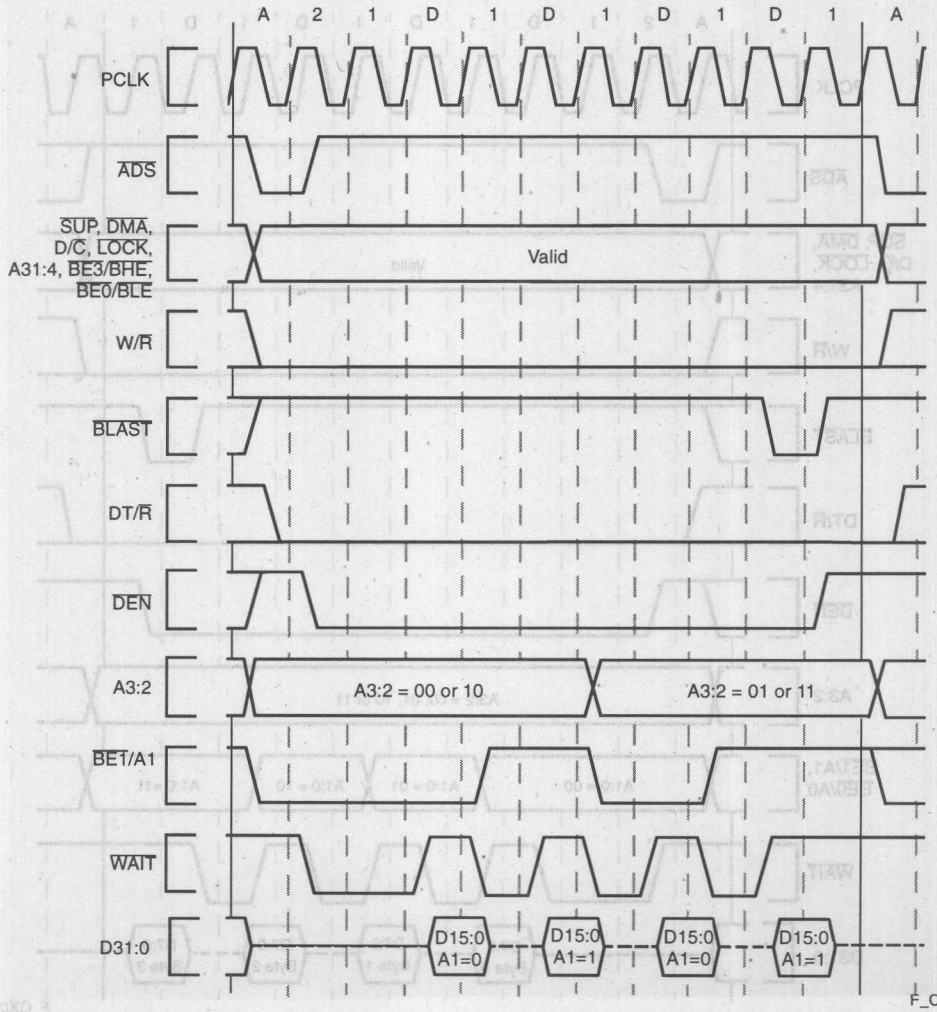


Figure 32. Burst, Non-Pipelined Read Request With Wait States, 16-Bit Bus

Function	reserved	Byte Order	reserved	Bus Width	N_{WDD}	N_{WAD}	N_{XDA}	N_{RDD}	N_{RAD}	Pipe-Lining	External Ready Control	Burst
Bit	31-23	22	21	20-19	18-17	16-12	11-10	9-8	7-3	2	1	0
Value	0 0..0	X x	0 0	8-bit 00	X xx	X xxxxx	1 01	1 01	2 00010	OFF 0	Disabled 0	Enabled 1

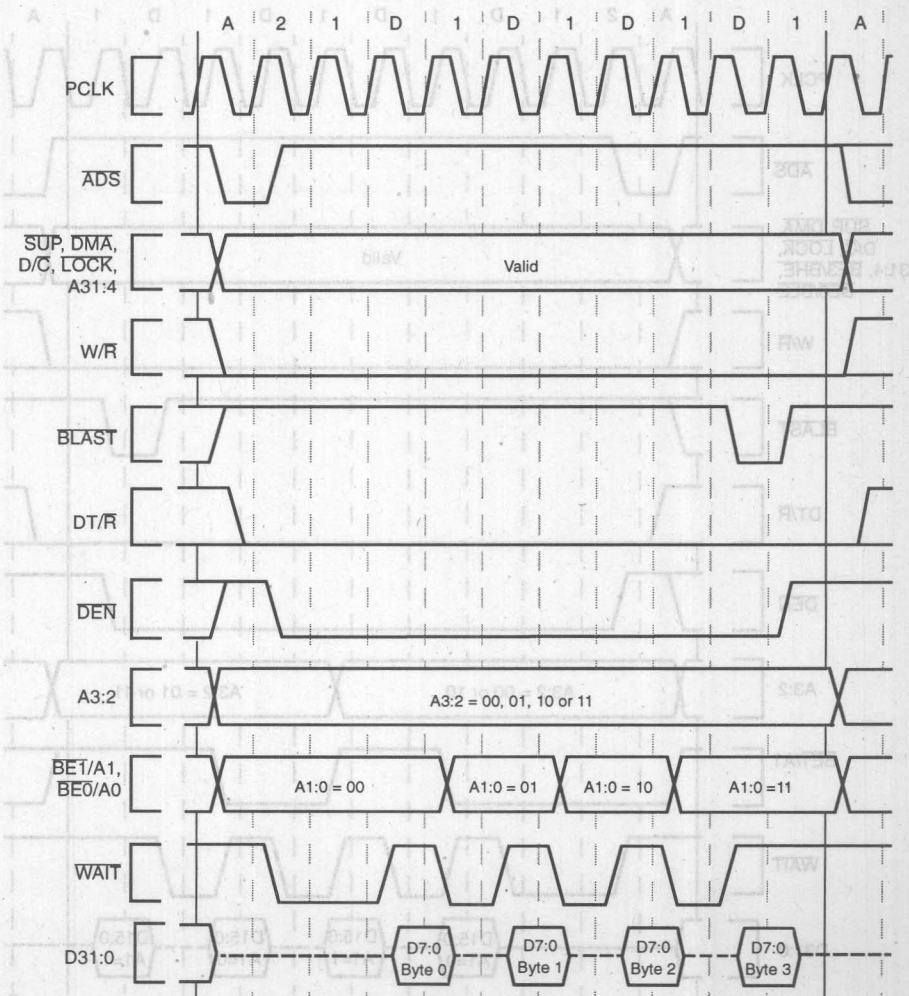


Figure 33. Burst, Non-Pipelined Read Request With Wait States, 8-Bit Bus

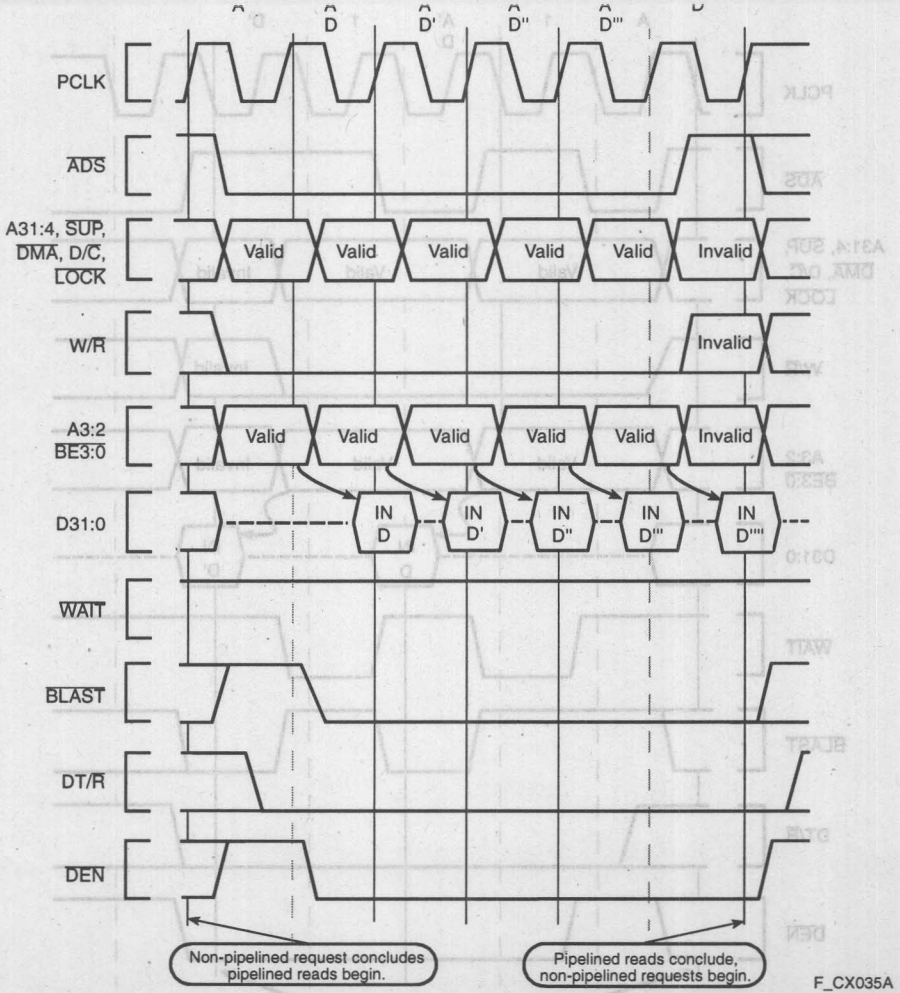


Figure 34. Non-Burst, Pipelined Read Request Without Wait States, 32-Bit Bus

Function	reserved	Byte Order	reserved	Bus Width	N_{WDD}	N_{WAD}	N_{XDA}	N_{RDD}	N_{RAD}	Pipe-Lining	External Ready Control	Burst
Bit	31-23	22	21	20-19	18-17	16-12	11-10	9-8	7-3	2	1	0
Value	0 0..0	X x	0 0	X xx	X xx	X xxxxx	X xx	X xx	1 00001	ON 1	X x	Disabled 1

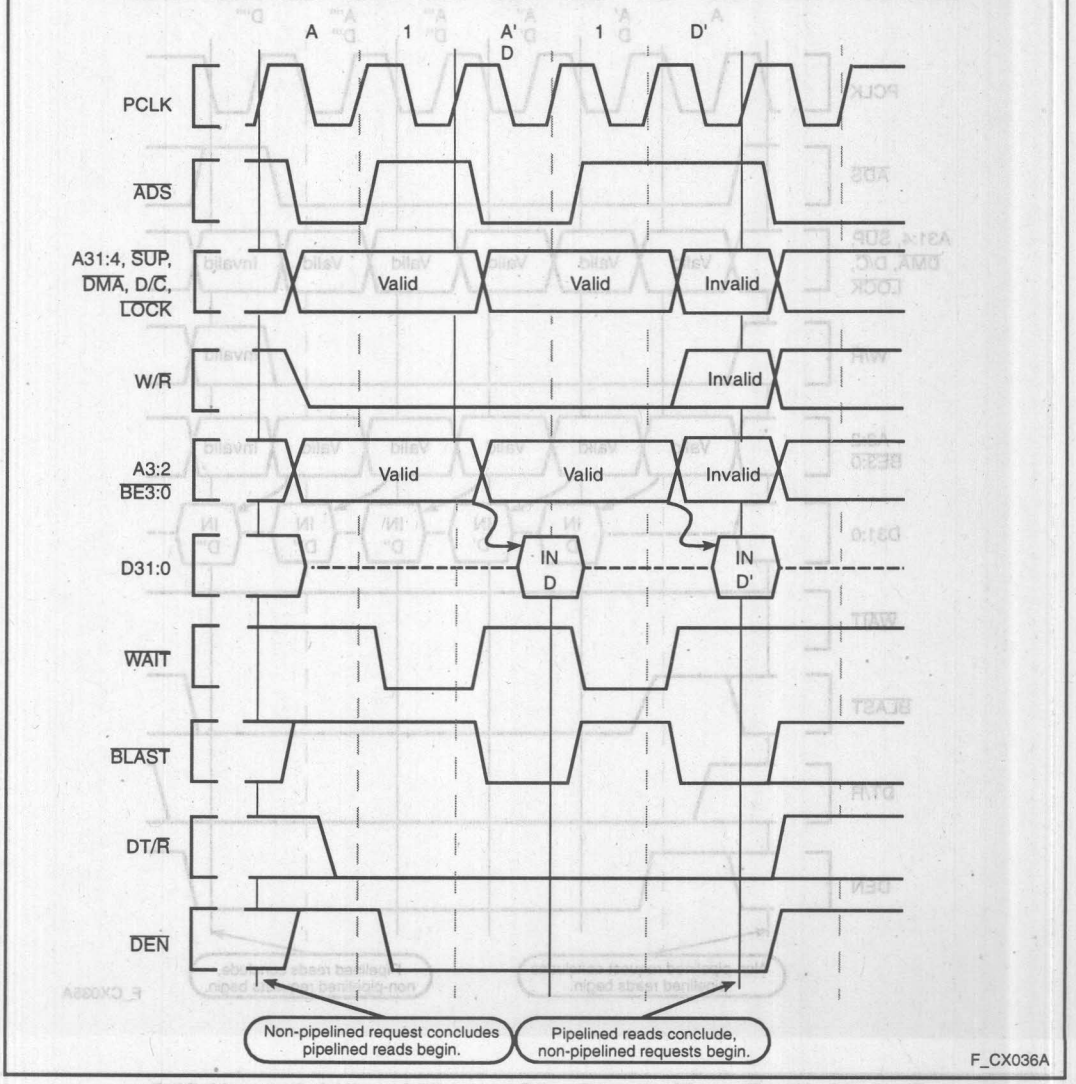


Figure 35. Non-Burst, Pipelined Read Request With Wait States, 32-Bit Bus

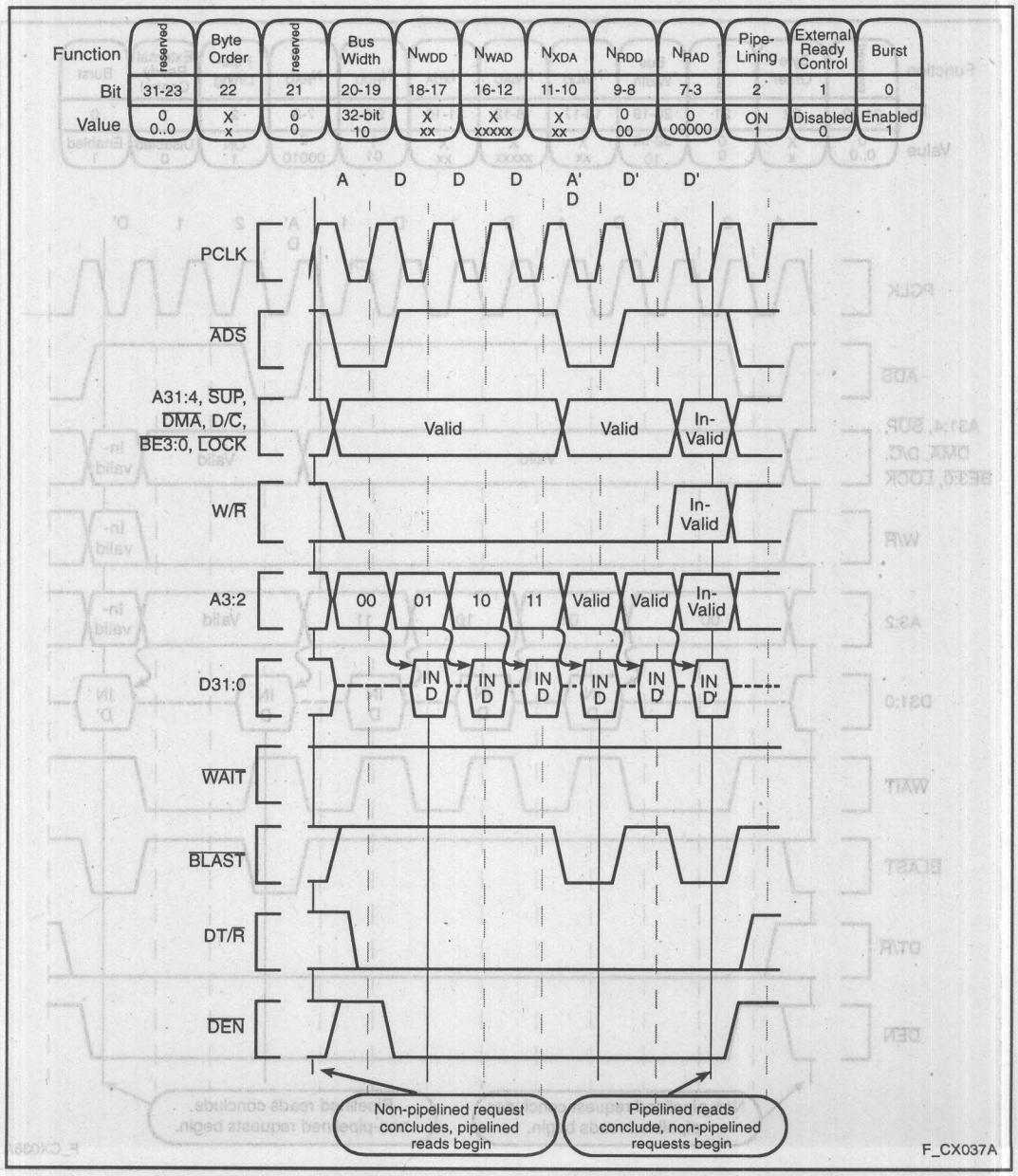


Figure 36. Burst, Pipelined Read Request Without Wait States, 32-Bit Bus

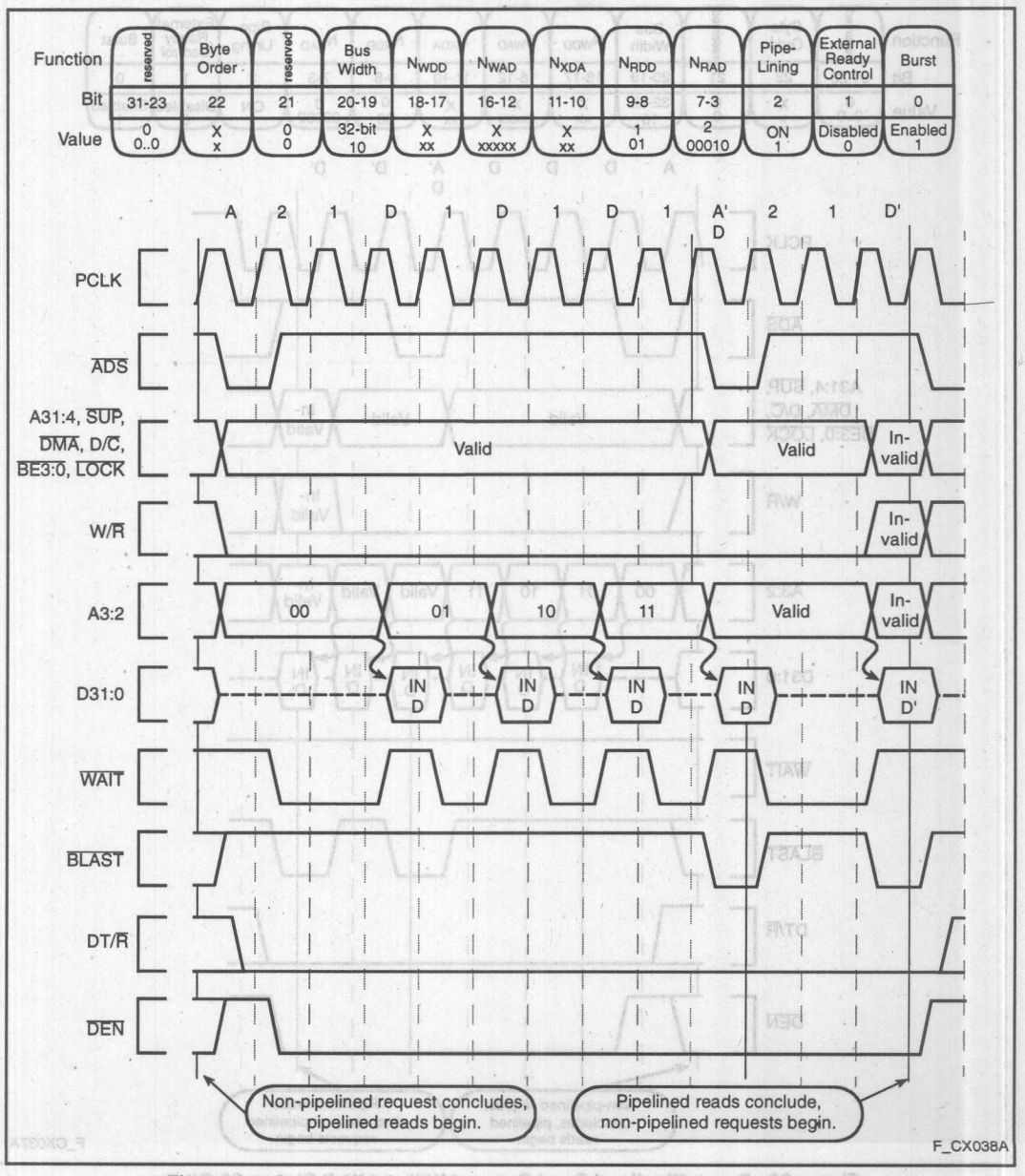


Figure 37. Burst, Pipelined Read Request With Wait States, 32-Bit Bus

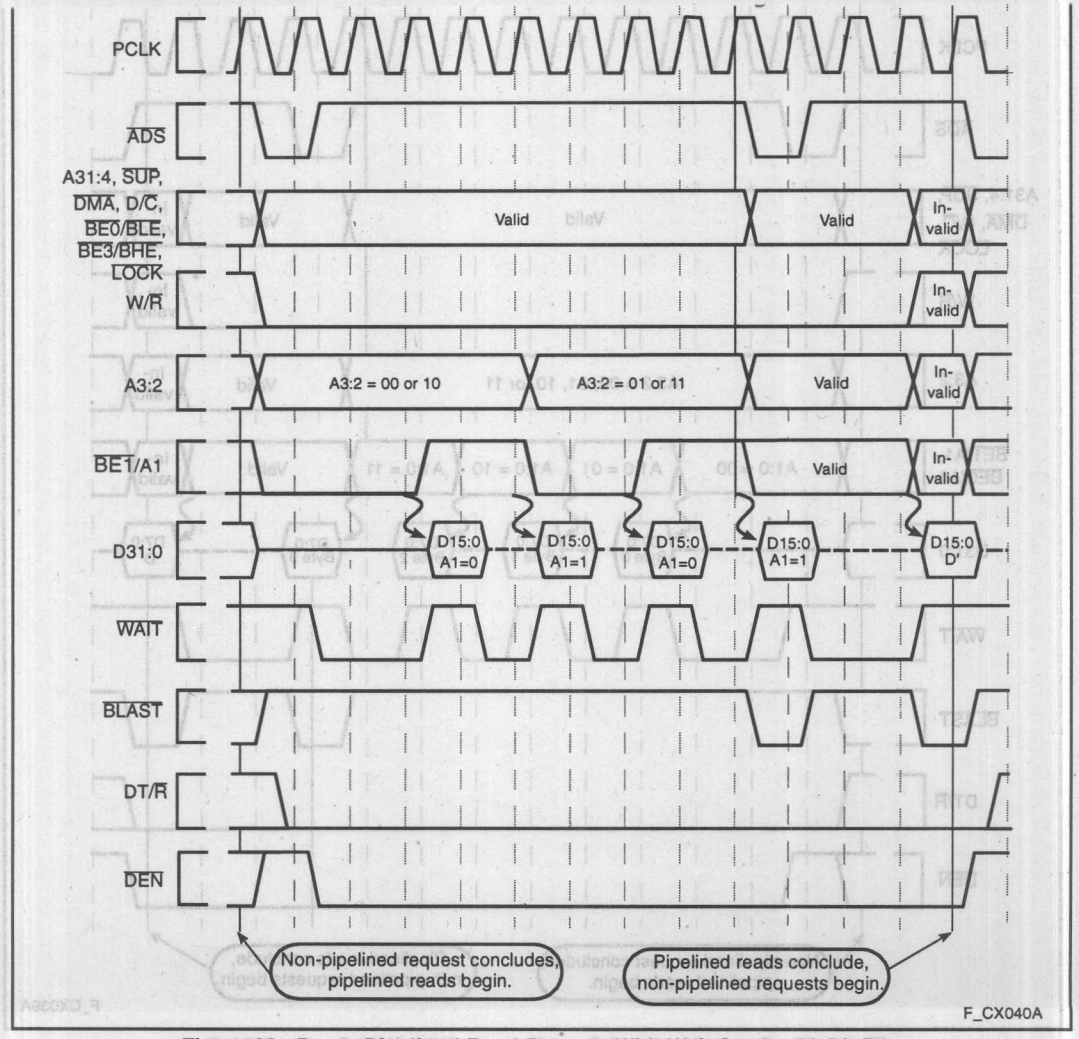


Figure 38. Burst, Pipelined Read Request With Wait States, 16-Bit Bus

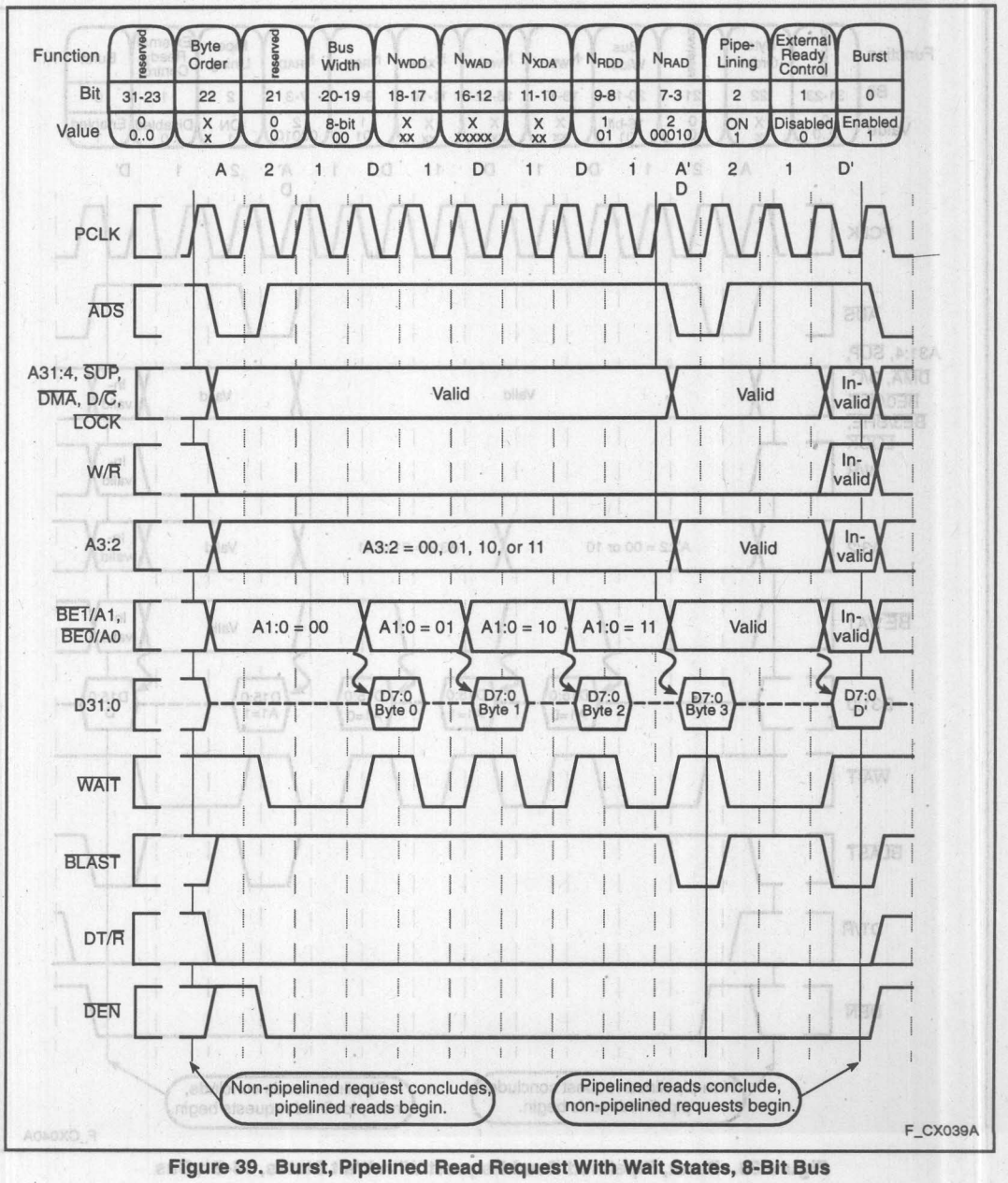


Figure 39. Burst, Pipelined Read Request With Wait States, 8-Bit Bus

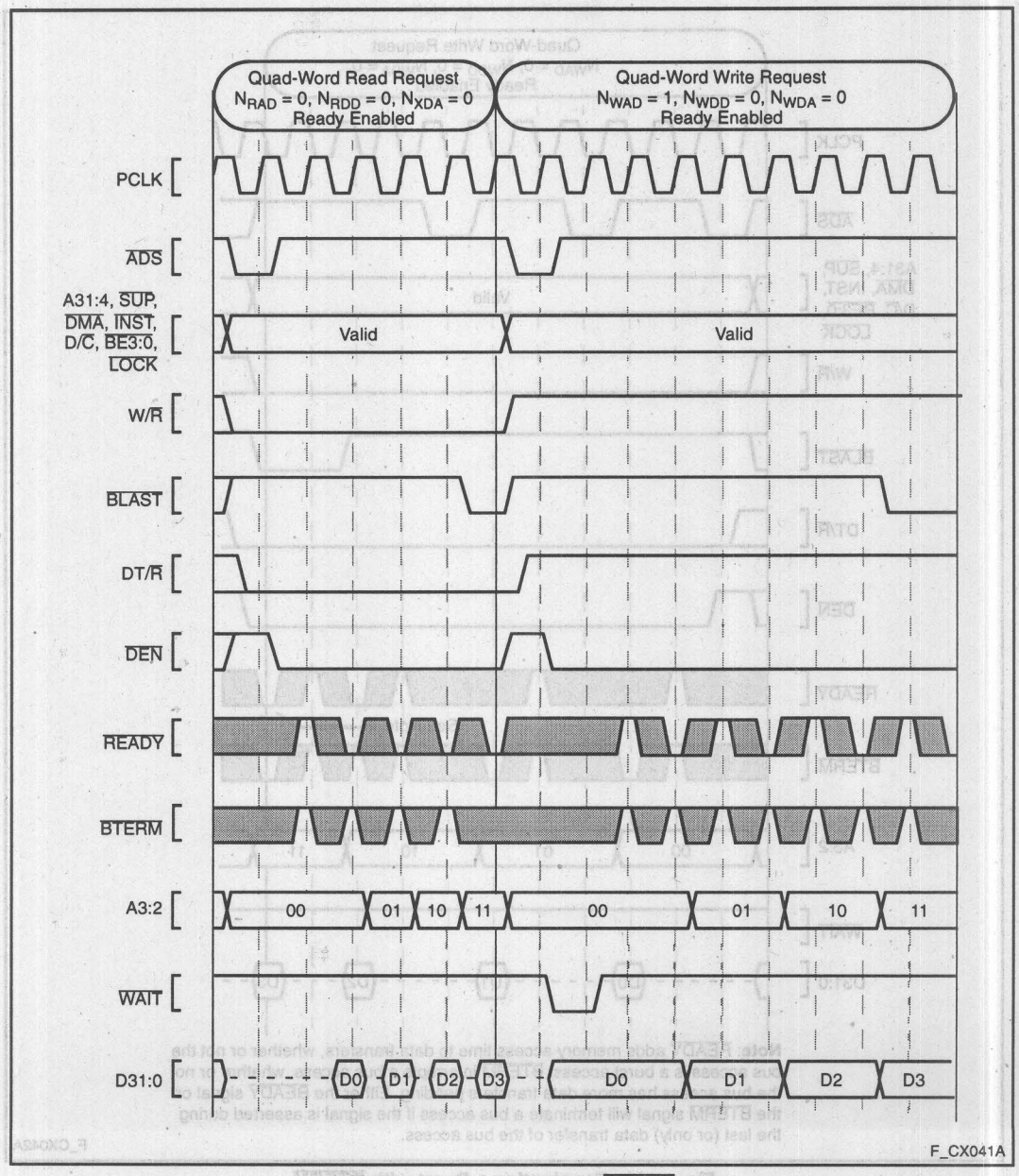


Figure 40. Using External READY

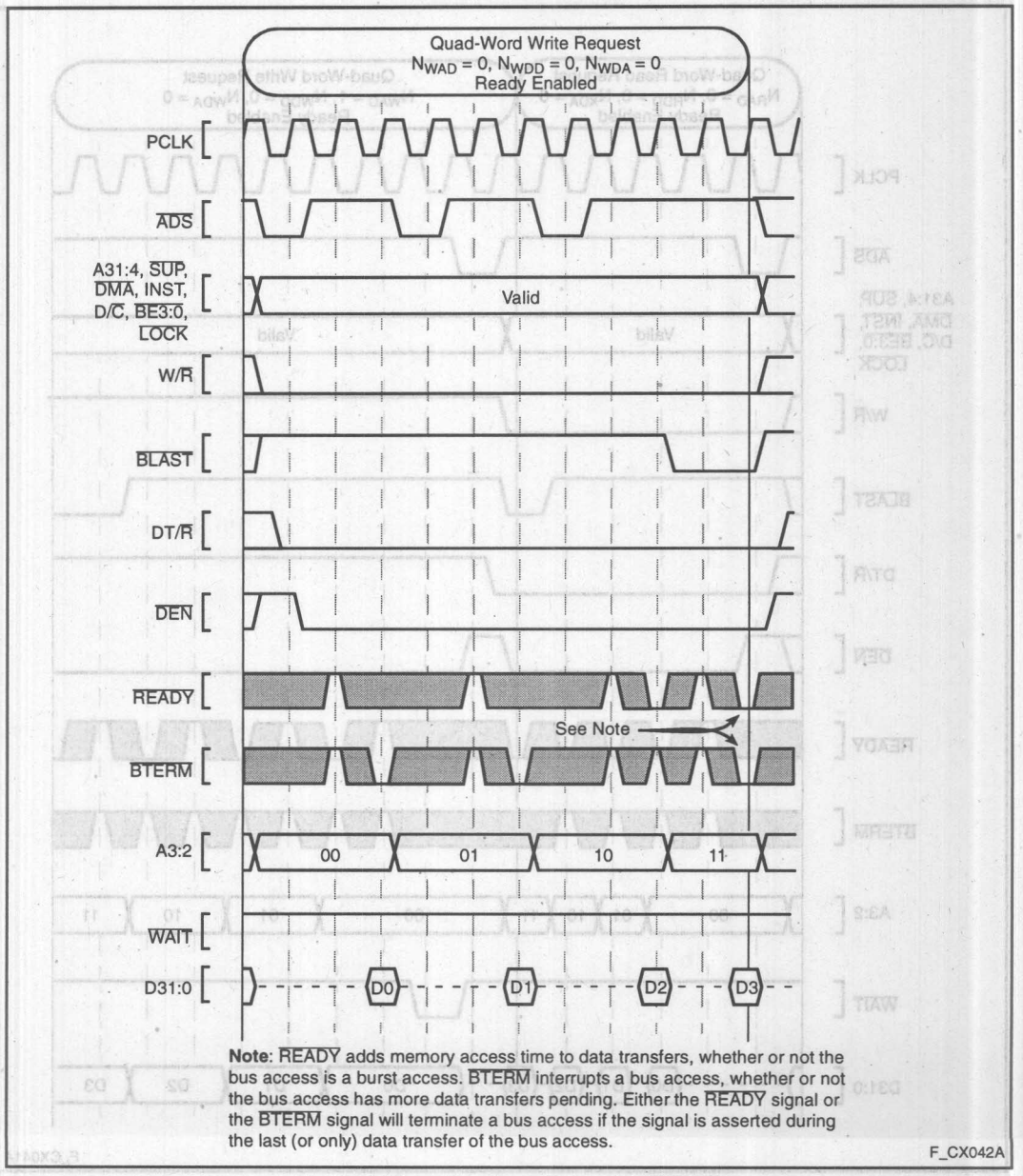


Figure 41. Terminating a Burst with BTERM

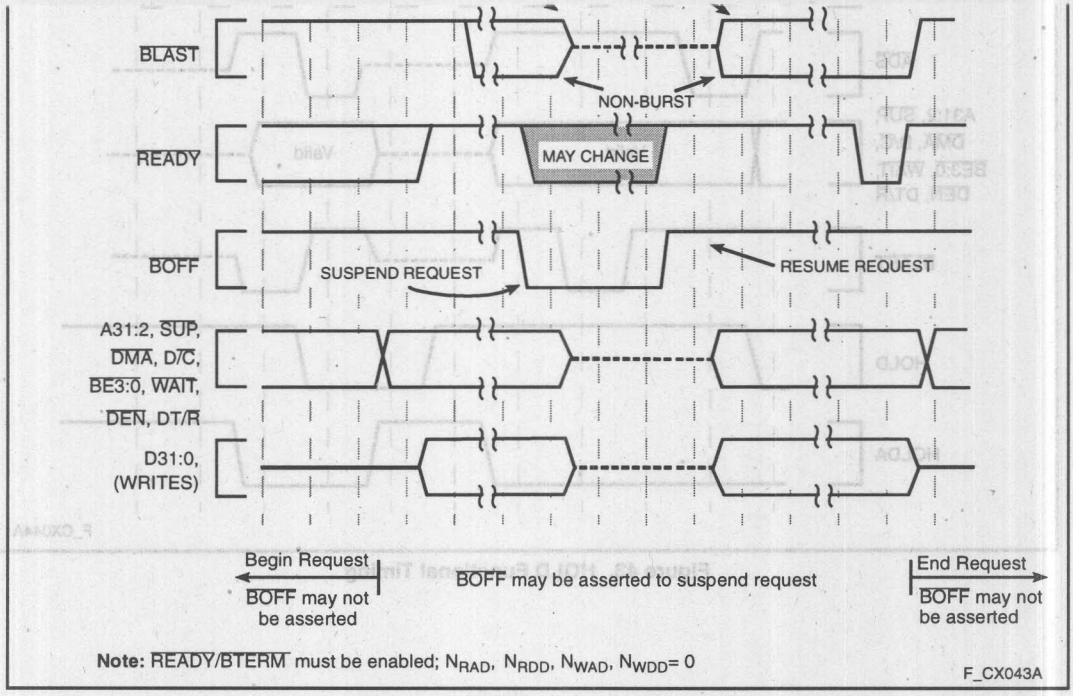


Figure 42. BOFF Functional Timing

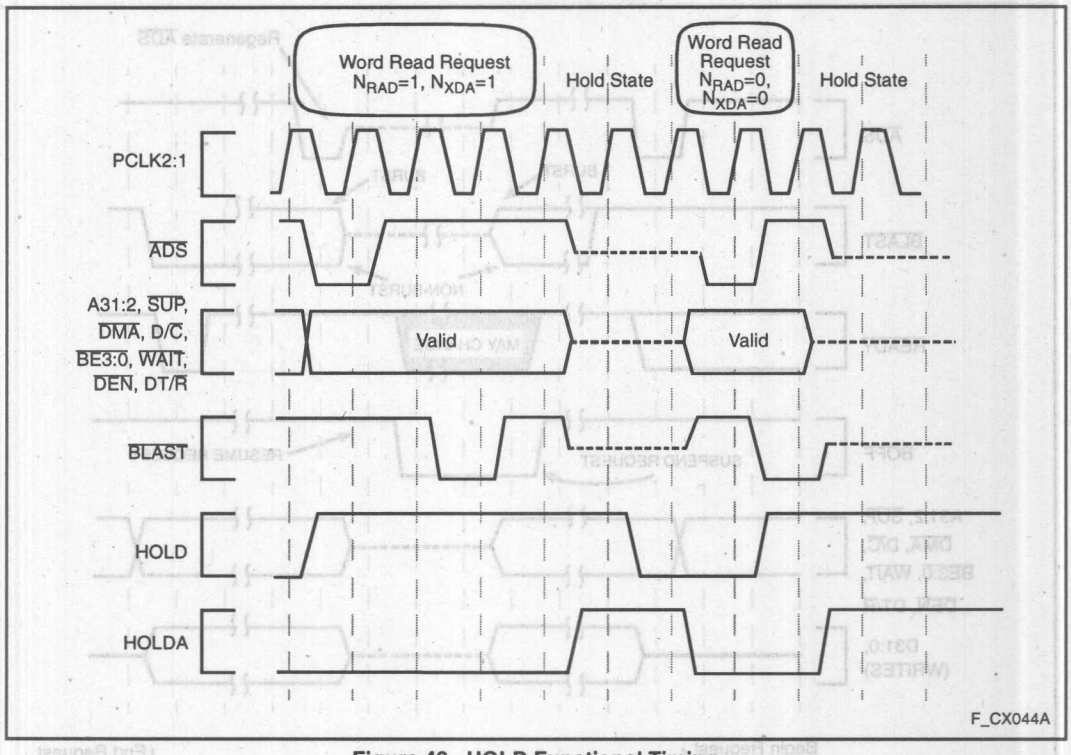
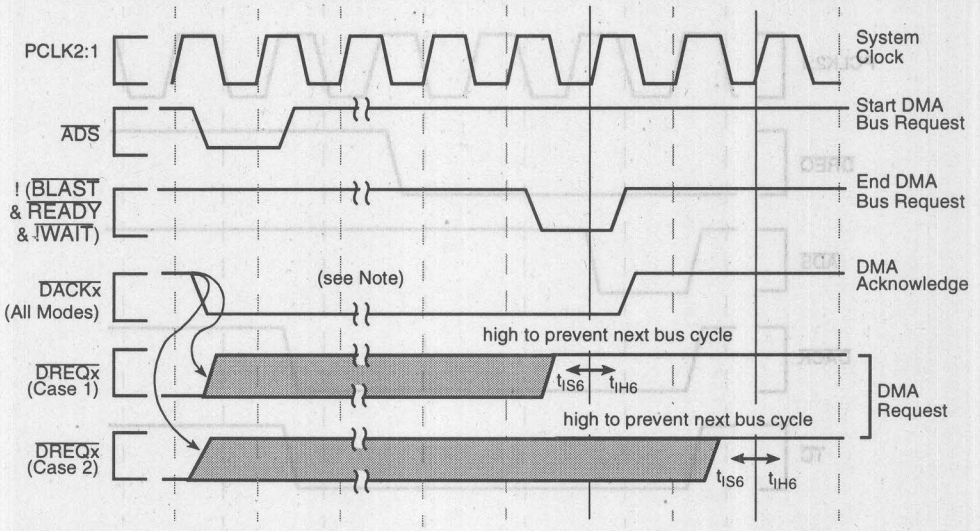


Figure 43. HOLD Functional Timing

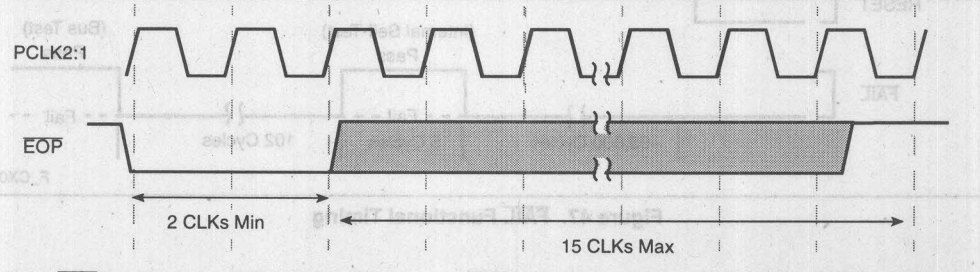


Note:

1. Case 1: DREQ must deassert before DACK deasserts. This applies to all Fly-By modes: source synchronized packing modes and destination synchronized unpacking modes.
2. Case 2: DREQ must be deasserted by the second clock (rising edge) after DACK is driven high. This applies to all other DMA transfers.
3. DACKx is asserted for the duration of a DMA bus request. The request may consist of multiple bus accesses (defined by ADS and BLAST).

F_CX018A

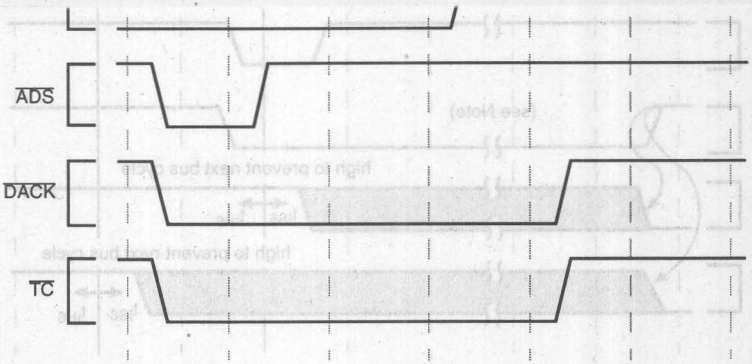
Figure 44. DREQ and DACK Functional Timing



Note: EOP has the same AC Timing Requirements as DREQ to prevent unwanted DMA requests. EOP is NOT edge triggered. EOP must be held for a minimum of 2 clock cycles then deasserted within 15 clock cycles.

F_CX045A

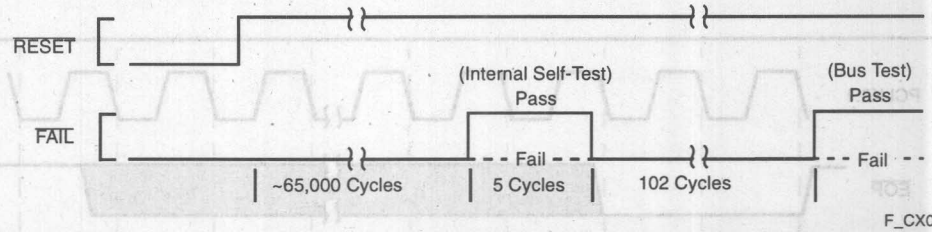
Figure 45. EOP Functional Timing



Note: Terminal Count becomes active during the last bus request of a buffer transfer. If the last LOAD/STORE bus request is executed as multiple bus accesses, the TC will be active for the entire bus request. Refer to the *i960® Cx Microprocessor User's Manual* for further information.

F_CX046A

Figure 46. Terminal Count Functional Timing



F_CX047A

Figure 47. FAIL Functional Timing

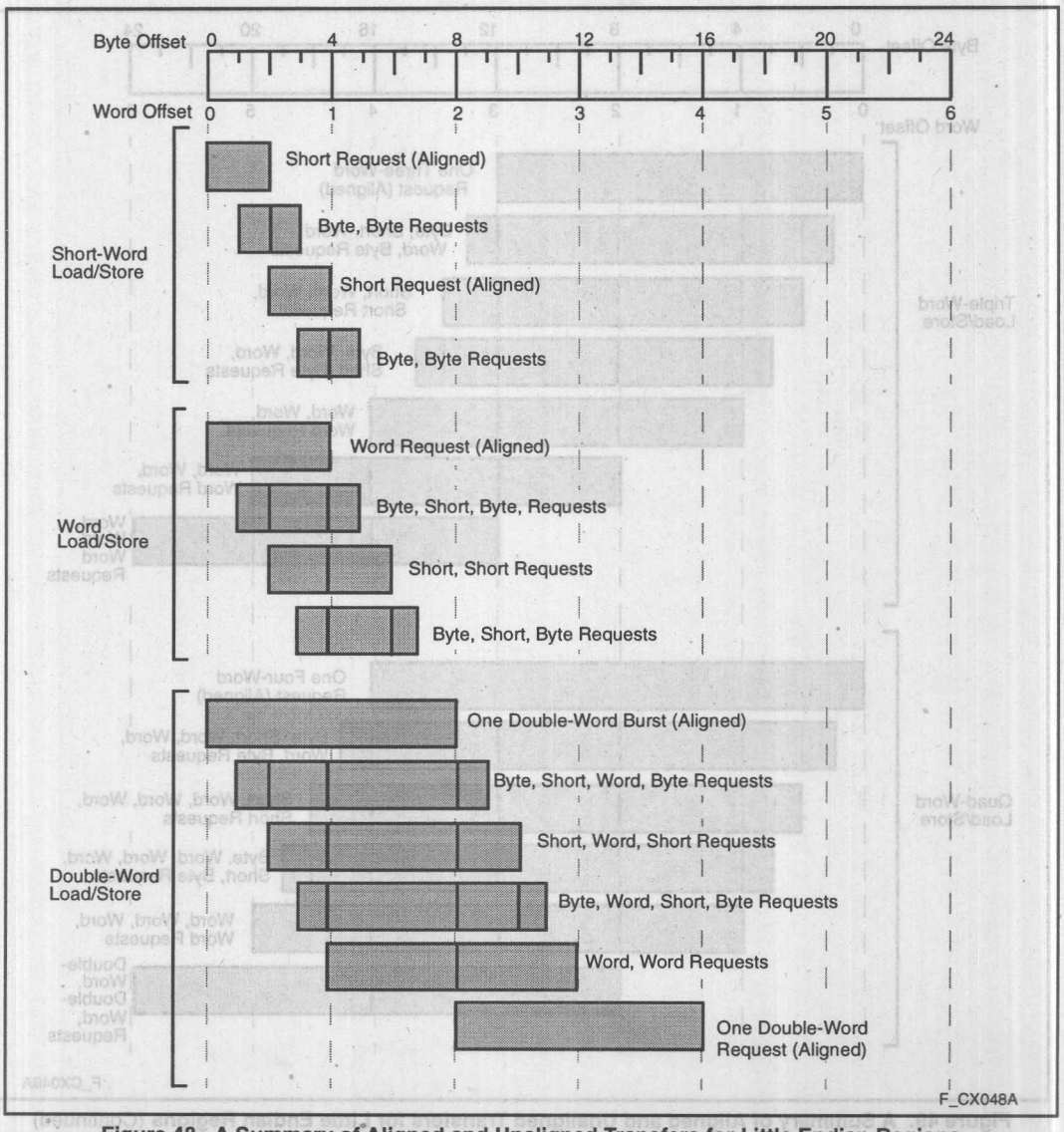


Figure 48. A Summary of Aligned and Unaligned Transfers for Little Endian Regions

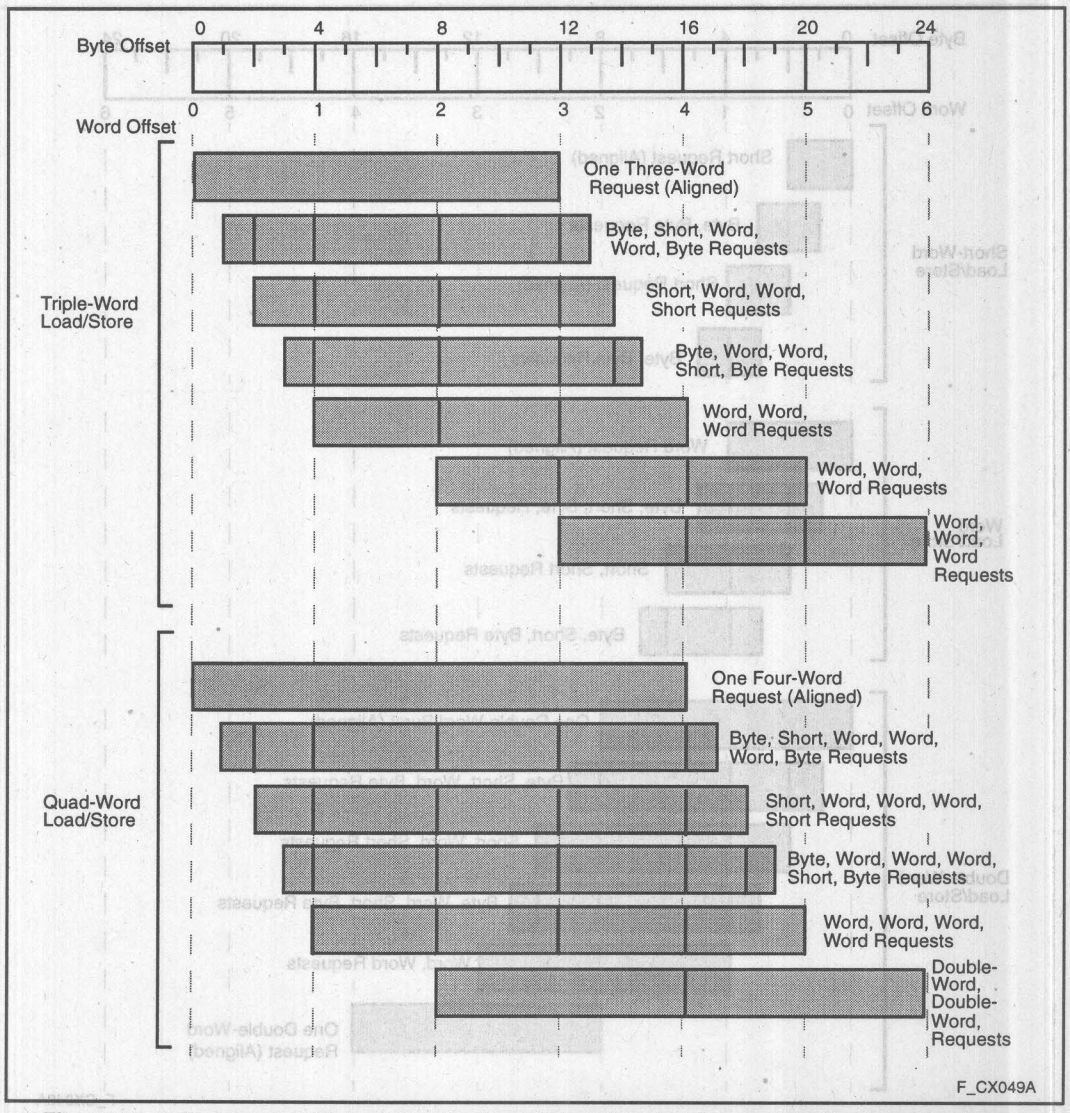


Figure 49. A Summary of Aligned and Unaligned Transfers for Little Endian Regions (Continued)

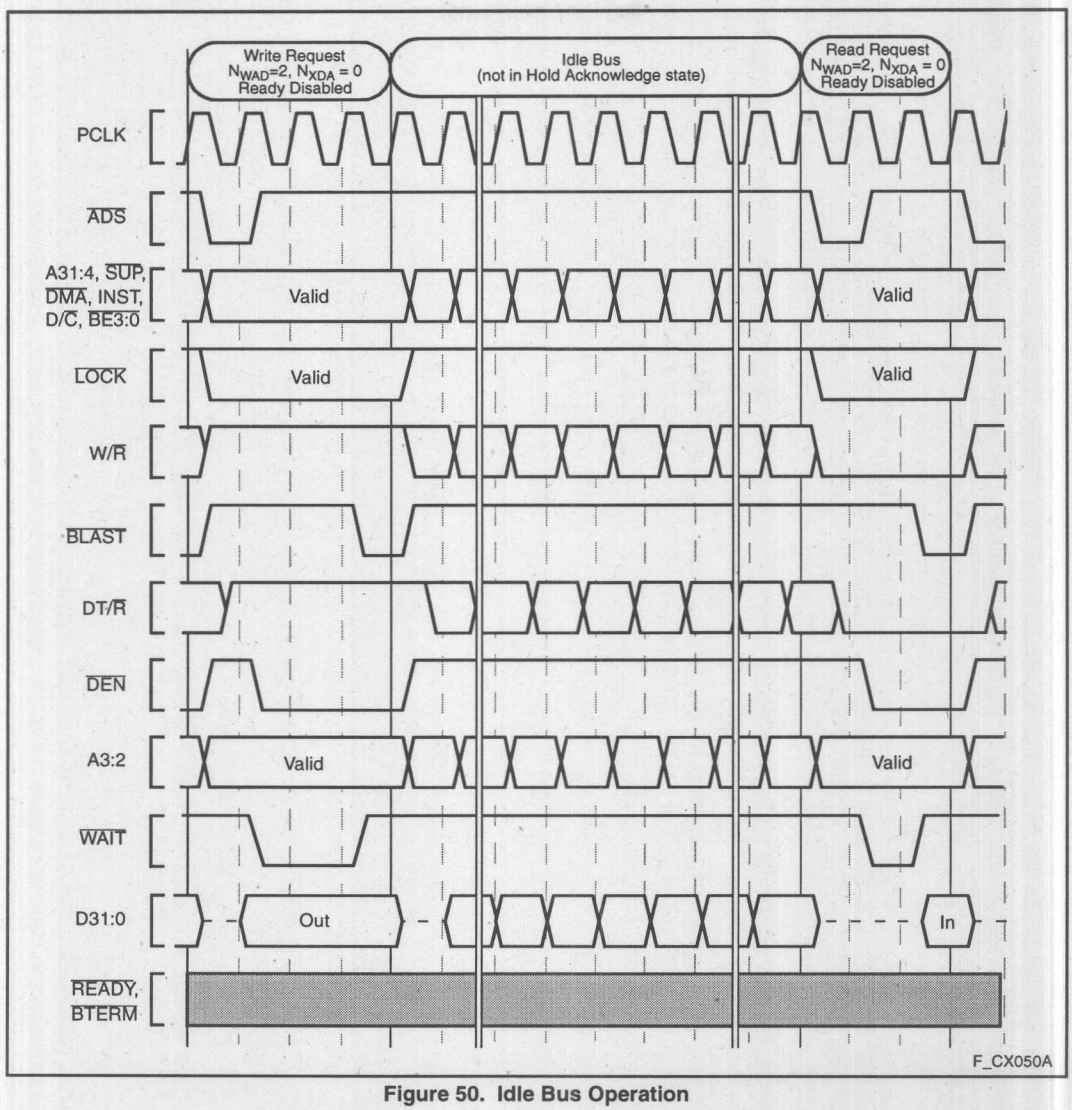


Figure 50. Idle Bus Operation

7.0 REVISION HISTORY

This is a new data sheet for the 80960CF-40 product. It is derived from the 80960CF-33, -25, -16 data sheet. Aside from a few minor edits, only the AC Characteristics differ from the 80960CF-33, -25, -16 data sheet.

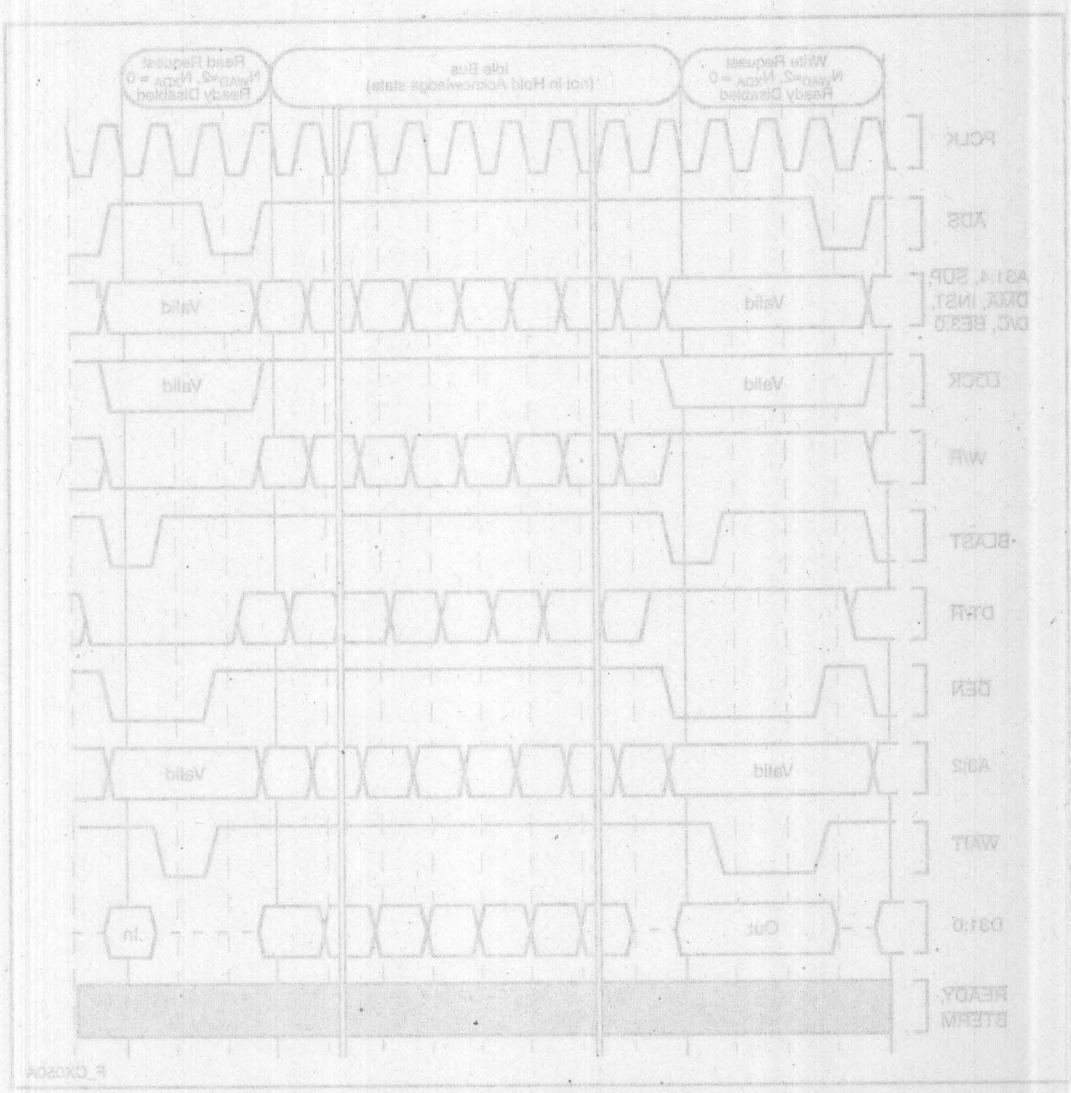


Figure 50. Idle Bus Operation

- — Direct Mapped
- — Parallel Load/Decode for Uncached Instructions
- **Multiple Register Sets**
  - — Sixteen Global 32-Bit Registers
  - — Sixteen Local 32-Bit Registers
  - — Four Local Register Sets Stored On-Chip
  - — Register Scoreboarding
- **4 Gigabyte, Linear Address Space**
- **Pin Compatible with 80960KB**
- — 66.7 Mbytes/s Burst
- — Up to 16 Bytes Transferred per Burst
- **132-Lead Packages:**
  - — Pin Grid Array (PGA)
  - — Plastic Quad Flat-Pack (PQFP)

The 80960KA is a member of Intel's i960® 32-bit processor family, which is designed especially for embedded applications. It includes a 512-byte instruction cache and a built-in interrupt controller. The 80960KA has a large register set, multiple parallel execution units and a high-bandwidth burst bus. Using advanced RISC technology, this high performance processor is capable of execution rates in excess of 9.4 million instructions per second*. The 80960KA is well-suited for a wide range of applications including non-impact printers, I/O control and specialty instrumentation.

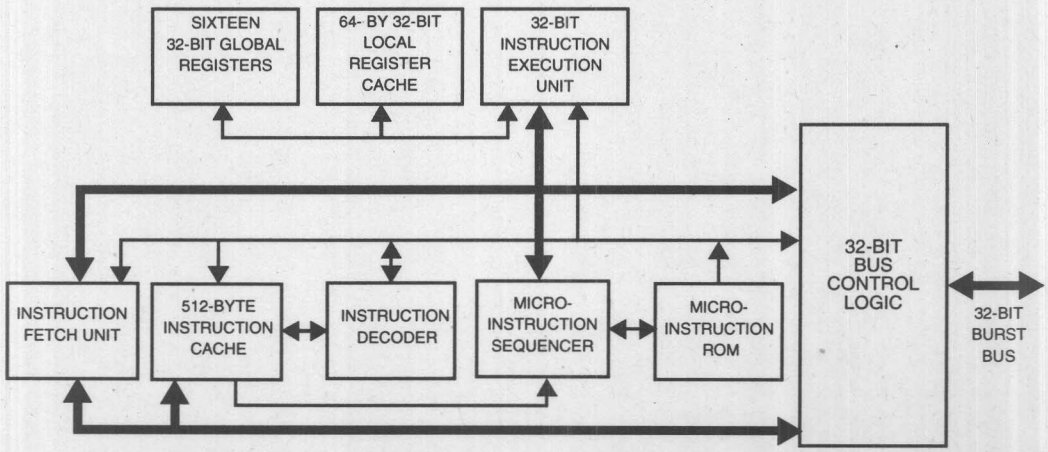


Figure 1. The 80960KA Processor's Highly Parallel Architecture

* Relative to Digital Equipment Corporation's VAX-11/780 at 1 MIPS (VAX-11™ is a trademark of Digital Equipment Corporation)

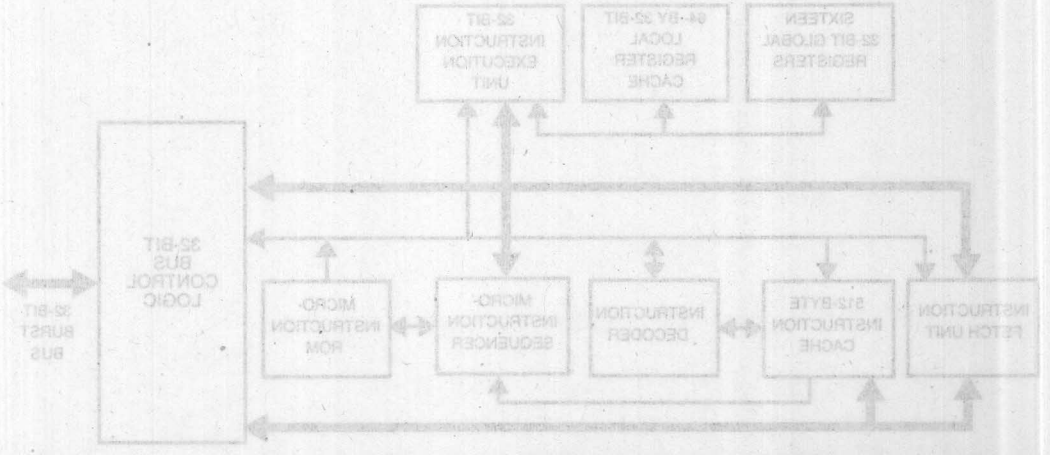
Order Number: 270775-005

This complete document is available from Intel's World Wide Website and/or U.S. Literature Center:

World Wide Website: http://www.intel.com

U.S. Literature Center : 800-548-4725

in other geographies, please contact your local sales office



80960KB
EMBEDDED 32-BIT MICROPROCESSOR WITH INTEGRATED FLOATING-POINT UNIT

- **High-Performance Embedded Architecture**
  - **— 25 MIPS Burst Execution at 25 MHz**
  - **— 9.4 MIPS* Sustained Execution at 25 MHz**
- **512-Byte On-Chip Instruction Cache**
  - **— Direct Mapped**
  - **— Parallel Load/Decode for Uncached Instructions**
- **Multiple Register Sets**
  - **— Sixteen Global 32-Bit Registers**
  - **— Sixteen Local 32-Bit Registers**
  - **— Four Local Register Sets Stored On-Chip**
  - **— Register Scoreboarding**
- **4 Gigabyte, Linear Address Space**
- **Pin Compatible with 80960KA**
- **Built-in Interrupt Controller**
  - **— 31 Priority Levels, 256 Vectors**
  - **— 3.4 µs Latency @ 25 MHz**
- **Easy to Use, High Bandwidth 32-Bit Bus**
  - **— 66.7 Mbytes/s Burst**
  - **— Up to 16 Bytes Transferred per Burst**
- **132-Lead Packages:**
  - **— Pin Grid Array (PGA)**
  - **— Plastic Quad Flat-Pack (PQFP)**
- **On-Chip Floating Point Unit**
  - **— Supports IEEE 754 Floating Point Standard**
  - **— Four 80-Bit Registers**
  - **— 13.6 Million Whetstones/s (Single Precision) at 25 MHz**

The 80960KB is a member of Intel's i960® 32-bit processor family, which is designed especially for embedded applications. It includes a 512-byte instruction cache, an integrated floating-point unit and a built-in interrupt controller. The 80960KB has a large register set, multiple parallel execution units and a high-bandwidth burst bus. Using advanced RISC technology, this high performance processor is capable of execution rates in excess of 9.4 million instructions per second*. The 80960KB is well-suited for a wide range of applications including non-impact printers, I/O control and specialty instrumentation.

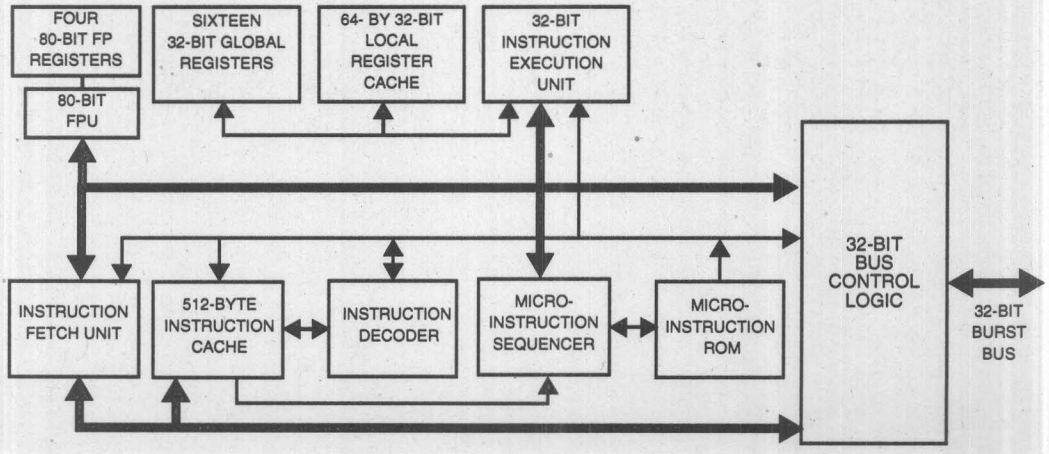


Figure 1. The 80960KB Processor's Highly Parallel Architecture

* Relative to Digital Equipment Corporation's VAX-11/780 at 1 MIPS (VAX-11™ is a trademark of Digital Equipment Corporation)

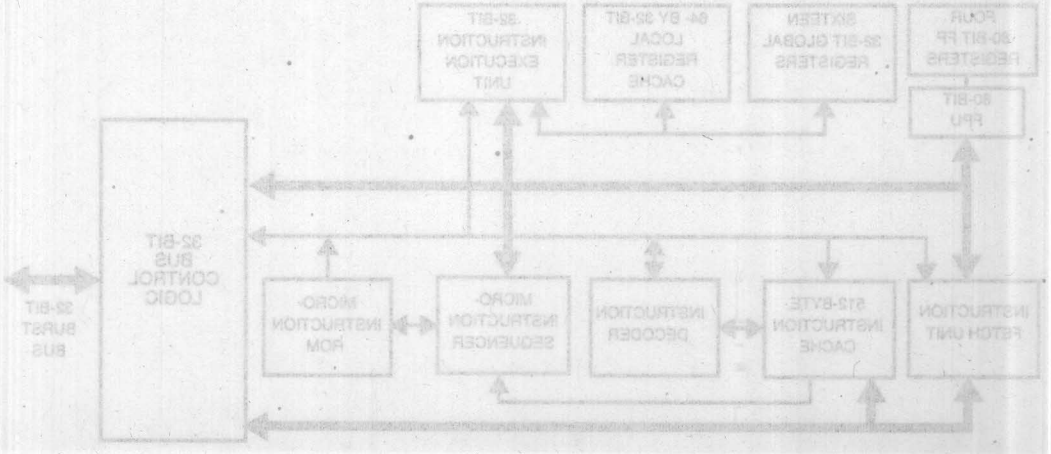
Order Number: 270565-006

This complete document is available from Intel's World Wide Website and/or U.S. Literature Center:

World Wide Website: http://www.intel.com

U.S. Literature Center : 800-548-4725

in other geographies, please contact your local sales office



- 20 MIPS* Burst Execution at 20 MHz
- 7.5 MIPS Sustained Execution at 20 MHz

- 512-Byte On-Chip Instruction Cache
  - Direct Mapped
  - Parallel Load/Decode for Uncached Instructions

- Multiple Register Sets
  - Sixteen Global 32-Bit Registers
  - Sixteen Local 32-Bit Registers
  - Four Local Register Sets Stored On-Chip
  - Register Scoreboarding

- Built-In Interrupt Controller
  - 4 Direct Interrupt Pins
  - 31 Priority Levels, 256 Vectors

- Easy to Use, High Bandwidth 16-Bit Bus
  - 32 Mbytes/s Burst
  - Up to 16 Bytes Transferred per Burst

- 32-Bit Address Space, 4 Gigabytes

- 80-Lead Quad Flat Pack (EIAJ QFP)
  - 84-Lead Plastic Leaded Chip Carrier (PLCC)

- Software Compatible with 80960KA/KB/CA/CF Processors

The 80960SA is a member of Intel's i960® 32-bit processor family, which is designed especially for low cost embedded applications. It includes a 512-byte instruction cache and a built-in interrupt controller. The 80960SA has a large register set, multiple parallel execution units and a 16-bit burst bus. Using advanced RISC technology, this high performance processor is capable of execution rates in excess of 7.5 million instructions per second*. The 80960SA is well-suited for a wide range of cost sensitive embedded applications including non-impact printers, network adapters and I/O controllers.

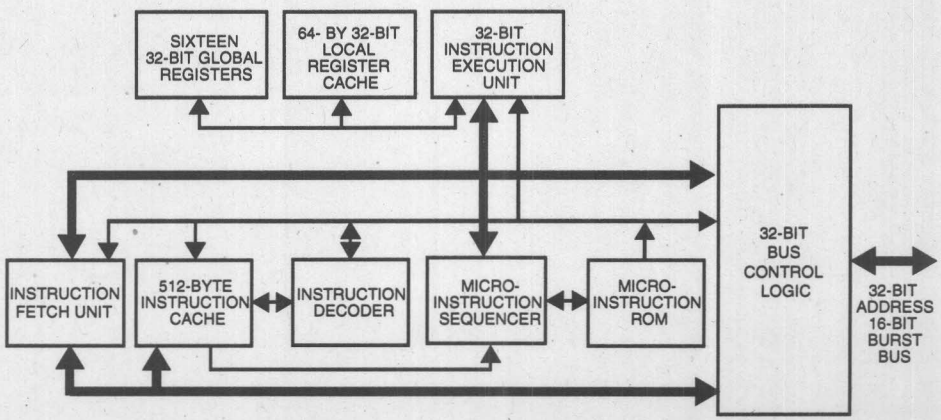


Figure 1. The 80960SA Processor's Highly Parallel Architecture

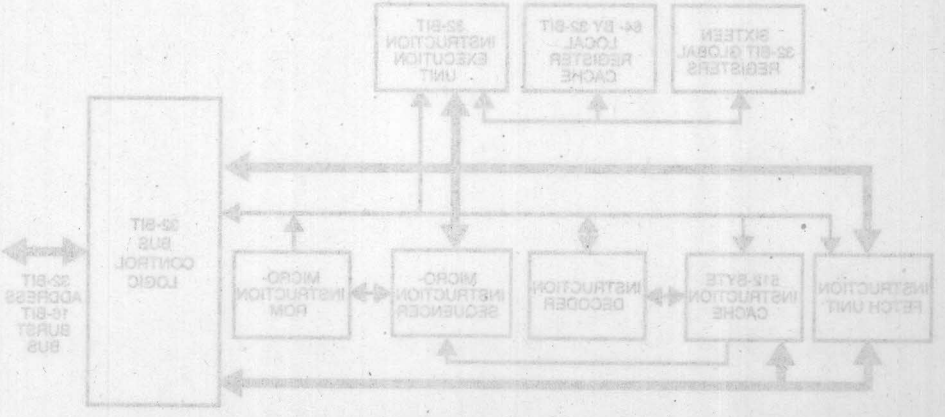
* Relative to Digital Equipment Corporation's VAX-11/780 at 1 MIPS (VAX-11™ is a trademark of Digital Equipment Corporation)

This complete document is available from Intel's World Wide Website and/or U.S. Literature Center:

World Wide Website: http://www.intel.com

U.S. Literature Center : 800-548-4725

in other geographies, please contact your local sales office



80960SB
EMBEDDED 32-BIT MICROPROCESSOR WITH 16-BIT BURST DATA BUS

- **High-Performance Embedded Architecture**
  - **16 MIPS* Burst Execution at 16 MHz**
  - **5 MIPS Sustained Execution at 16 MHz**
- **512-Byte On-Chip Instruction Cache**
  - **Direct Mapped**
  - **Parallel Load/Decode for Uncached Instructions**
- **Multiple Register Sets**
  - **Sixteen Global 32-Bit Registers**
  - **Sixteen Local 32-Bit Registers**
  - **Four Local Register Sets Stored On-Chip**
  - **Register Scoreboarding**
- **Pin Compatible with 80960SA**
- **Built-in Interrupt Controller**
  - **4 Direct Interrupt Pins**
  - **31 Priority Levels, 256 Vectors**
- **Built-In Floating Point Unit**
  - **Fully IEEE 754 Compatible**
- **Easy to Use, High Bandwidth 16-Bit Bus**
  - **25.6 Mbytes/s Burst**
  - **Up to 16 Bytes Transferred per Burst**
- **32-Bit Address Space, 4 Gigabytes**
- **80-Lead Quad Flat Pack (EIAJ QFP)**
  - **84-Lead Plastic Leaded Chip Carrier (PLCC)**
- **Software Compatible with 80960KA/KB/CA/CF Processors**

The 80960SB is a member of Intel's i960® 32-bit processor family, which is designed especially for low cost embedded applications. It includes a 512-byte instruction cache, an integrated floating-point unit and a built-in interrupt controller. The 80960SB has a large register set, multiple parallel execution units and a 16-bit burst bus. Using advanced RISC technology, this high performance processor is capable of execution rates in excess of 5 million instructions per second*. The 80960SB is well-suited for a wide range of cost sensitive embedded applications including non-impact printers, network adapters and I/O controllers.

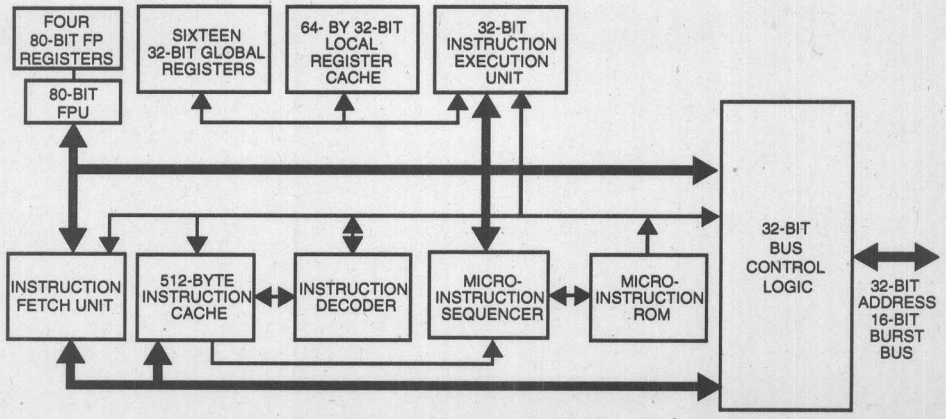


Figure 1. The 80960SB Processor's Highly Parallel Architecture

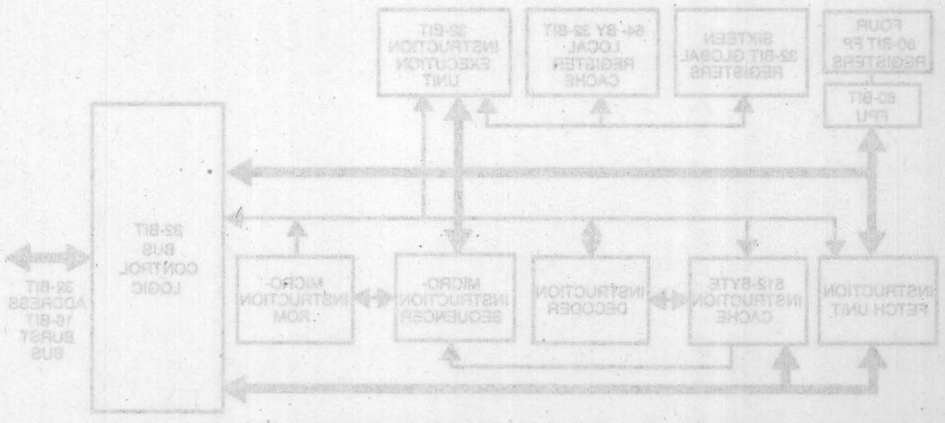
* Relative to Digital Equipment Corporation's VAX-11/780 at 1 MIPS (VAX-11™ is a trademark of Digital Equipment Corporation)

This complete document is available from Intel's World Wide Website and/or U.S. Literature Center:

World Wide Website: http://www.intel.com

U.S. Literature Center : 800-548-4725

in other geographies, please contact your local sales office



18 October 1995

Dear Customer:

Thank you for ordering the Applications Note AP-506, *Designing for 80960Cx and 80960Hx Compatibility*, which describes how to build a board to accept either the i960® Cx and the Hx processors.

Since we published Applications Note AP-506, the Data Sheet AC specifications have been revised based on design simulations. Also, some errata have been discovered in the Applications Note. The enclosed sheet describes updates for AP-506.

Please contact either one of us if you have any questions.

Sincerely,

Mike Pulley
80960Hx Applications Engineer
Mike_J_Pulley@ccm.hf.intel.com

Jasmine Yamasaki
80960Hx Product Marketing
Jasmine_B_Yamasaki@ccm.hf.intel.com

Update #1: Page 14

Was:

- A Hx outputs data within a 7 ns window -- between 1.5 and 8.5 ns.

Should be:

- For a 5V I/O system, the Hx produces data within an **11.5** ns window -- between 1.5 and **13** ns.
- For a 3.3V I/O system, the Hx produces data within an **8.5** ns window -- between 1.5 and **10** ns.

Update #2: Table on Page 14

Was:

		DO(80960Cx)	DO(80960Hx)	Combined
Output timing	TOV (max)	16 ns	8.5 ns	16 ns
Input timing	TIS (min)	3 ns	5 ns	5 ns

Should be:

		DO(80960Cx)	DO(80960Hx)	Combined
Output timing				
5V I/O	TOV (max)	16 ns	**13** ns	16 ns
3.3V I/O	**TOV (max)**	**n/a**	**10 ns**	**16 ns**
Input timing	TIS (min)	3 ns	**6** ns	**6** ns

Update #3: Table 4. 80960Cx/80960Hx Pin Differences *on Page 16*

Was:

Pin	CA/CF	Hx	80960Hx-ready System
A6	DREQ1#	TRST#	...This pin should be pulled high when not in use.

Should be:

Pin	CA/CF	Hx	80960Hx-ready System
A6	DREQ1#	TRST#	...This pin should be pulled **low** when not in use.

Update #4: Add the following section to Page 11

FAIL PIN

Many applications use a light emitting diode (LED) to indicate when the FAIL# pin is low (active). However, when an Hx processor is in the socket, and the FAIL# pin is high (inactive) at 3.3V, the LED can still be forward biased enough to glow. To ensure the LED extinguishes when FAIL# goes high on both processors, Intel recommends the circuit shown in Figure 8, below.

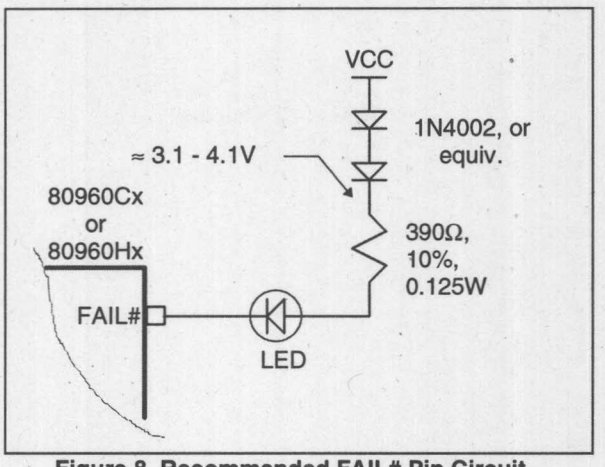


Figure 8. Recommended FAIL# Pin Circuit

The two diodes dissipate about 1.4V, so the LED voltage drops too low to glow when the Hx FAIL# pin goes high. Use a low current LED that can operate at 3-5 mA. This design works whether VCC is 5V or 3.3V nominal.

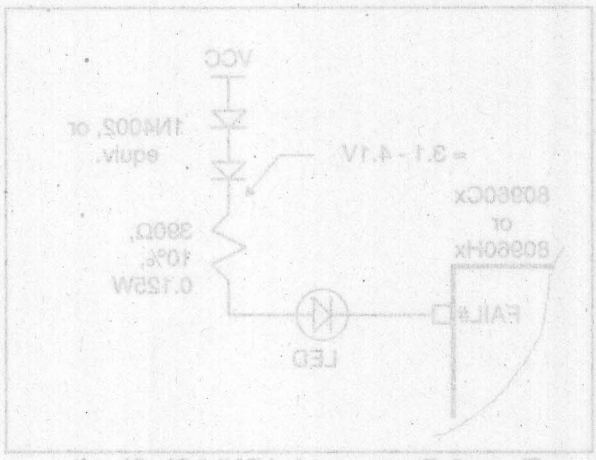
An alternative is to eliminate the diodes and power the LED from a 3.3V VCC supply for the Hx and from a 5V VCC supply for the Cx.

Update #5: Add the following bullet to the Design Guideline Summary on Page 17

- Reduce the voltage on the FAIL# LED.

Update #6: Add the following paragraph to the Reserved Memory section on Page 13

Some systems use controller chips that map their control registers into memory region 0xff000000 to 0xffffffff. Since the Hx processor does not access external addresses within that region, those systems should externally invert address bit 24 to the controller chip to move the registers to 0xfe000000 to 0xfeffffff.


VCC
1N4002, or
equiv.
= 3.1 - 4.1V
80960Cx
or
80960Hx
390Ω,
10%,
0.125W
FAIL#
LED

Designing for 80960Cx and 80960Hx Compatibility

March 1995

Designing for 80960Cx and 80960Hx Compatibility

March 1995

INTRODUCTION

The 80960HA/HD/HT[1] processors, Intel's new superscalar i960® processor, adds new features and performance to the other well-known products in the i960 processor family. The 80960Hx is designed to satisfy the compute-intensive, data throughput performance requirements of both today's applications and those of the future.

This document addresses the important hardware considerations when designing a "80960Hx ready" system[2]. This is a system which is designed to use an i960 Cx processor[3] and can also use the 80960Hx processor (when available). To help simplify this task, pinout for the 80960Hx PGA package is similar to pinout for the i960 Cx processor PGA package. Although the 80960Hx is not drop-in compatible with all Cx designs, systems can be built with a CPU socket footprint which will accept either processor.

A summary of the most important hardware design considerations are:

power supply — V_{CC} for the Cx is 5V; the Hx uses 3.3V. An 80960Hx-ready system's power supply must accommodate these voltage requirements.

DMA controller — Cx processors have a built-in DMA controller; the Hx does not. An 80960Hx-ready system should not use the Cx built-in DMA controller. Cx pins used for DMA control have different function on the Hx.

byte enable signals — The Hx's byte enable encodings are a superset of the Cx byte enable encodings. The 80960Hx-ready system should be designed to accept all combinations of byte enable encodings.

bus arbitration — The Hx does not grant HOLD requests during an atomic operation (assert HOLDA in response to HOLD), but Cx processors will grant HOLD

Endnotes

1 Throughout this document, "Hx" refers to the i960 HA, HD and HT processors. Information that is specific to each is clearly indicated.

2 "80960Hx-ready" refers to a system designed to use a CA/CF processor that can also use an 80960Hx.

3 Throughout this document, "Cx" refers to both the i960 CA and CF processors. Information that is specific to each is clearly indicated.

requests after any bus request, including in the middle of atomic accesses. A 80960Hx-ready system must not allow HOLD requests when the external LOCK pin is asserted if semaphore operations are to be performed between bus masters.

The Hx has an additional arbitration signal — BSTALL — which can be used by an external arbiter to indicate the processor has stalled because the bus controller is busy. (The Cx does not have BSTALL.)

external interrupts Interrupt subsystems must produce asynchronous interrupt inputs. The Hx samples interrupts differently than the Cx processors.

N_{XDA} wait states A system must not rely on N_{XDA} wait states between each access. Although both the Hx and Cx processors have programmable N_{XDA} wait states, behavior in the Hx is different. The Hx always inserts N_{XDA} wait states between accesses. The Cx only inserts N_{XDA} wait states between bus "requests." Each bus request can cause multiple bus accesses.

An 80960Hx-ready system must NOT accept data on writes during N_{XDA} wait states. During N_{XDA} wait states, the Hx processor drives the D31:0 bus. Cx processors do not drive valid data during N_{XDA} wait states.

parity The Hx provides built-in byte parity; Cx processors do not. If parity is used when the system contains an Hx processor, pull-up resistors must be provided to ensure that inputs sent to either the processor or to the external parity system do not float.

boundary scan The Hx has an IEEE 1149.1 JTAG interface; conversely, the Cx does not support JTAG. If JTAG is used when the system contains a Cx processor, the processor must be externally bypassed in the JTAG chain.

reserved memory Accesses to reserved memory (0xffxxxxxx) do not appear on the Hx bus. The Cx uses 0xffffffxx to fetch the Initial Boot Record. External decoders should map this memory to two different areas in the processor's address space.

AC timing AC specifications differ for Hx and Cx processors. Of course, AC timing analysis must be performed when designing a 80960Hx-ready system. The Cx AC timings are referenced to PCLK2:1; on the Hx, AC timings are referenced to CLKIN. (The Hx does not have PCLK2:1 signals.)

POWER REQUIREMENTS

The Hx requires a V_{cc} of 3.3V while the Cx operate at 5V. A system can be designed with a socket that accepts either processor. The Hx processor may be damaged if plugged into a socket that supplies 5V V_{cc}. Jumpers, switches, programmable power regulators, or other V_{cc} switching must be provided to select the proper V_{cc} for the processor. The 80960Hx's VOLDET pin can be used to accommodate automatic voltage selection circuitry.

An 80960Hx-ready system requires 5V on the VCC5 pin to provide 5V tolerant inputs.

Providing 3.3 V in a 5 V System

In most system board designs, the 5 V system power supply is routed to the components on the board through a dedicated board layer. With the requirement of a new 3.3 V supply for the Hx, it is not necessary to add a completely new power supply layer to the circuit board, as it is possible to create a 3.3 V "island" around the processor in the existing power supply plane.

Figure 1 shows a recommended "island" layout. The Hx processor's 5 V tolerant input buffers and TTL compatible outputs allow the processor to interface with existing TTL compatible external logic without requiring extra components. Thus, the processor can run at 3.3 V while the system logic runs at 5 V.

Other important considerations are:

- The "island" needs to be large enough to include the processor, the required power supply decoupling capacitance, and the necessary connection to the 3.3 V source.
- To minimize signal degradation, the gap between the 3.3 V "island" and the 5 V plane should be kept small. A typical gap size is about 0.02 inches.
- Minimize the number of traces routed across the power plane gap, since each crossing introduces signal degradation due to the impedance discontinuity that occurs at the gap. For traces that must cross the gap, route them on the side of the board next to the ground plane to reduce or eliminate the signal degradation caused by crossing the gap. If this is not possible, route the trace to cross the gap at a right angle (90 degrees).
- Use liberal decoupling capaciatnce between the 5V plane and the 3.3V island. A 0.01 µf ceramic capacitor every 0.5 to 1.0 inches along the perimeter of the island will greatly reduce the impedance discontinuity.

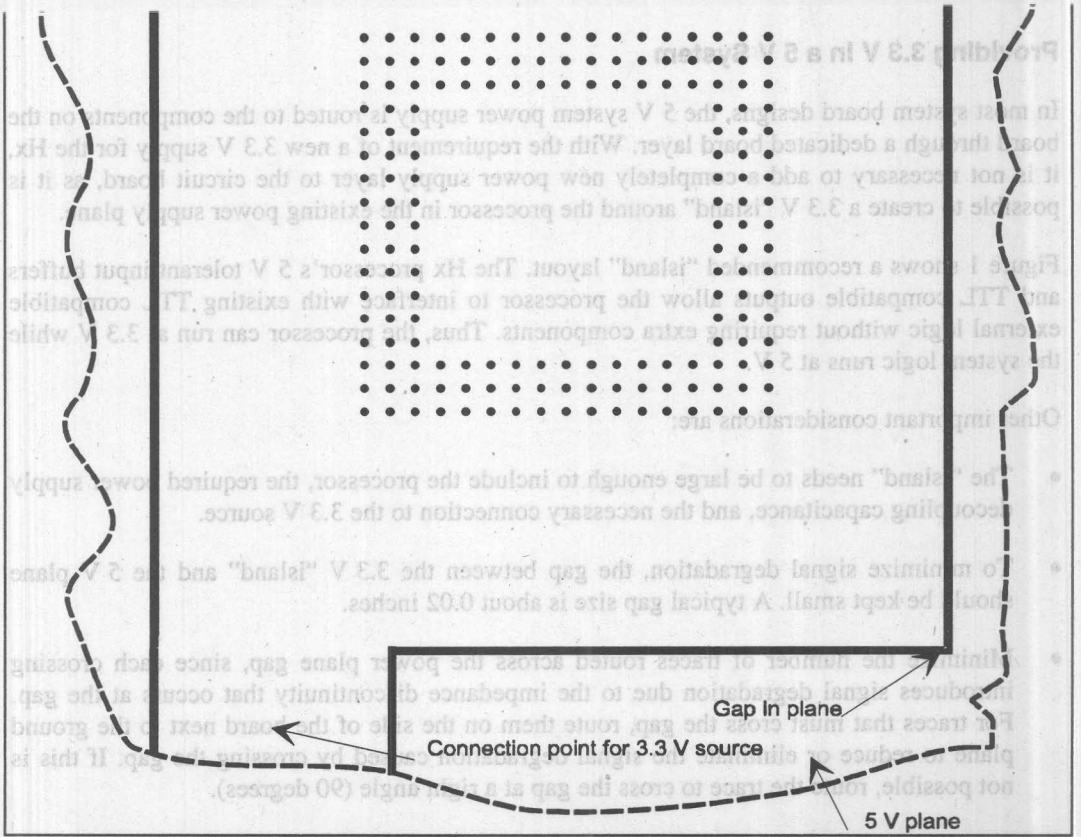


Figure 1. Creating a Power "Island"

Choosing a Power Source

The primary concerns which must be addressed when selecting a power source are maximum and minimum load current requirements and response time. The processor power supply must be able to maintain correct voltage regulation at current levels below 10 mA for the Hx in the HALT Mode, and up to the maximum current of 1.5 A.

Executing a HALT instruction causes the Hx to enter the HALT Mode, which causes a significant reduction in the current consumption of the processor in as few as 100 ns. The transition from HALT to the Normal State causes current consumption to return to the normal levels in a similarly short period of time. The processor power supply must be able to maintain correct voltage regulation during these transitions.

There are basically two options for supplying 3.3 V to the processor, either:

- Add a 3.3 V tap to the primary system power supply
- Use on-board secondary regulation to derive 3.3 V from the 5 V system power supply

For on-board secondary regulation, a linear voltage regulator will perform adequately for most designs. If low heat or power dissipation is a design goal, the higher complexity and cost of a switching regulator may be warranted. Switching regulators offer better efficiency, thereby lowering regulator power consumption and heat.

Figure 2 shows recommended layouts for power supply or linear regulator connection to the 3.3 V "island."

Power Supply Selection For Flexible Systems

Using the 80960Hx's voltage detect sense feature, you may design a flexible system which will automatically provide the proper processor voltage for an 80960Hx or Cx processor. It is also possible to make the selection of processor voltage an option during system board assembly.

VOLDET Automatic Voltage Select Circuit Option

By sampling the VOLDET pin at powerup, system boards can automatically select the processor power supply voltage, enabling a design that may use the 3.3V Hx or a 5 V Cx processor without jumpers or assembly time changes. The VOLDET pin is only present in the PGA package version of the Hx. This pin, which is an NC (No Connect) on the Cx processor, is connected internally to V_{SS} on the Hx. This pin should be left unconnected in designs that do not use the voltage detect feature.

Figure 3 shows an example of VOLDET pin usage with a linear regulator circuit to automatically select the correct power supply voltage. If VOLDET is not connected inside the processor, indicating a 5 V part, the gate of MOSFET Q1 is pulled high, which bypasses the 3.3 V regulator, supplying 5 V directly to the processor. Shorting the regulator's input to the output in this way is harmless for most linear regulators, due to regulator feedback circuitry which shuts the regulator off (contact regulator manufacturers for specifics). Note that in this case, most regulators require Q1 to handle all the processor's current requirements, and so should be a high-current, low on-state-resistance MOSFET. If VOLDET is connected to V_{SS}, indicating a 3.3 V part, the Q1 transistor is turned off, allowing the regulator to function normally. Figure 4 shows a suggested placement and layout for MOSFET Q1.

Other Voltage Selection Options

It is also possible to design a flexible system board where the processor supply voltage is selected by an assembly time option. There are several methods to achieve this; the key requirement being that the design must handle the maximum current of 1.5 A.

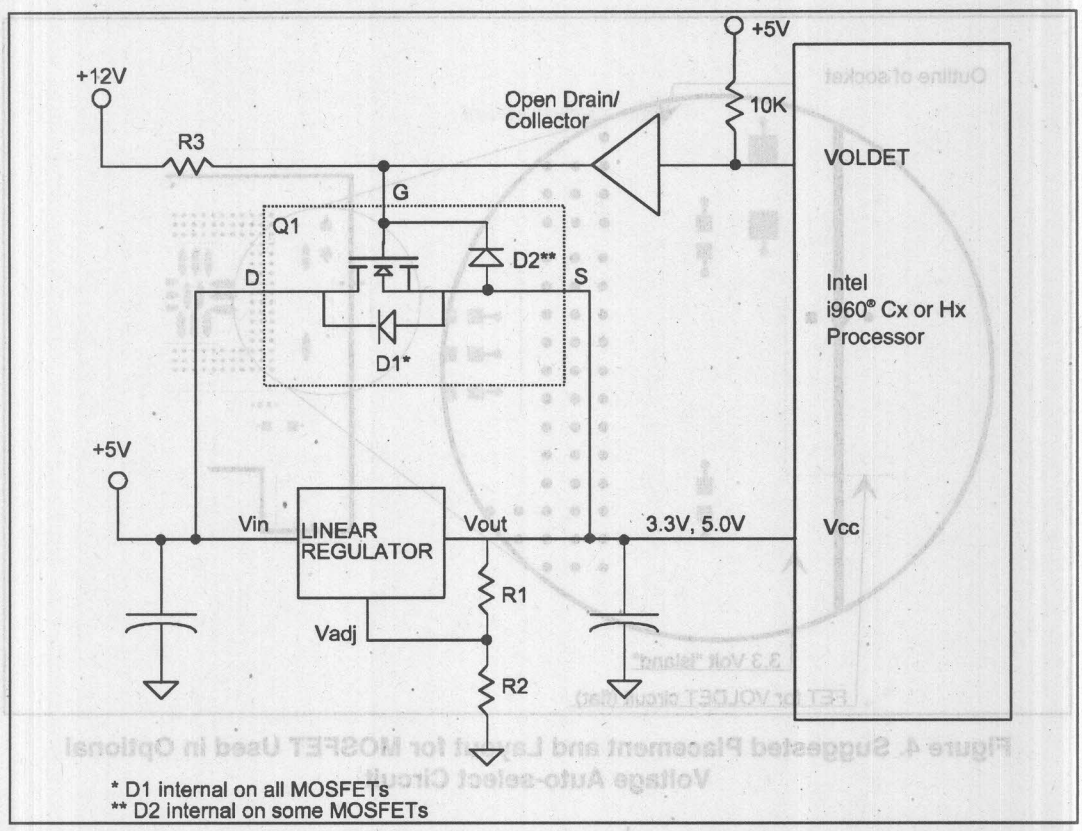


Figure 3. Example Voltage Auto-Select Circuit Topology[4]

Endnotes

[4] Illustration courtesy of Linear Technology Corporation

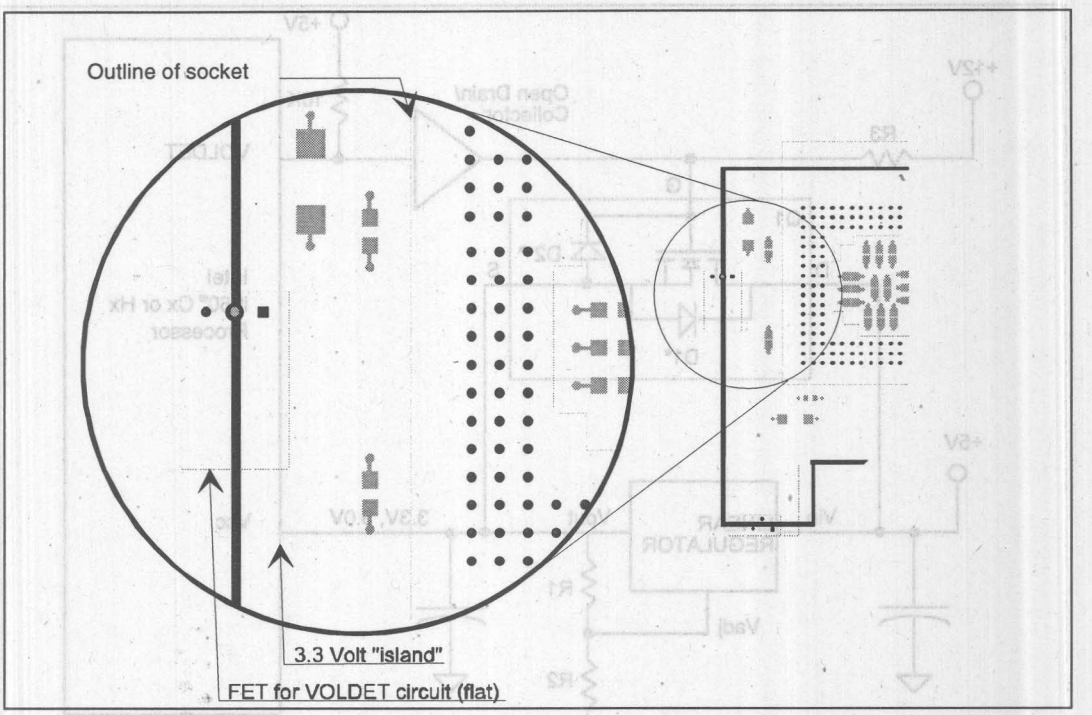


Figure 4. Suggested Placement and Layout for MOSFET Used in Optional Voltage Auto-select Circuit

VCC5 Pin Requirement

For mixed voltage systems where the processor interfaces with 5 V components, the VCC5 pin must be connected to 5 V for proper 5 V tolerant buffer operation. **The VCC5 input should not exceed VCC by more than 2.25 V during power-up, power-down or during operation.** If this requirement is not met, current flow through the pin may exceed 55 mA which may damage the component. To meet this requirement, one of two things must be done:

- The power supply must be designed to turn on and off such that the difference between the VCC5 and VCC voltages never exceeds 2.25 V, or,
- A 100 Ω resistor must be put in series with the VCC5 pin to limit the current through this path (Figure 5 shows a possible layout for this connection).

The 100 Ω series resistor is required for power supplies which do not meet the voltage difference specification, and also provides protection in the case of a power supply failure (where the 5 V supply remains on, but the 3.3 V supply goes to zero).

The VCC5 pin corresponds to a NC (no connect) pin on the Cx processor. This pin has no effect on the operation of the Cx, and can be driven.

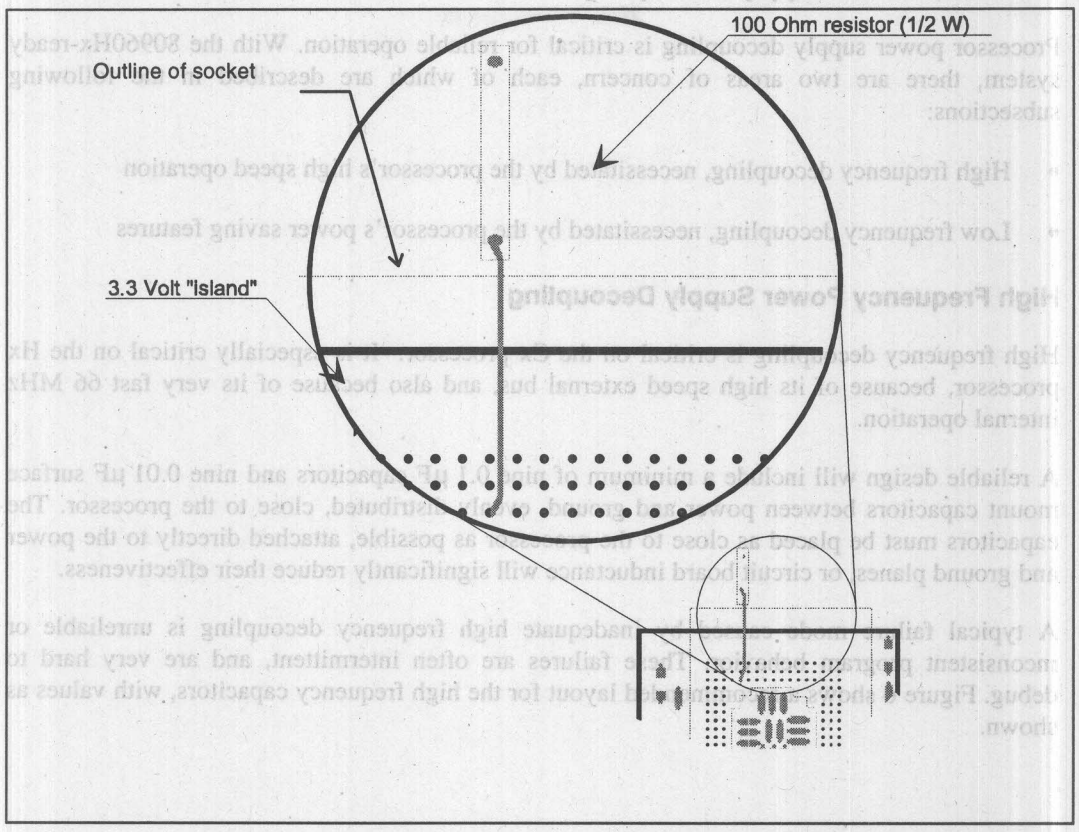


Figure 5. Possible Layout For VCC5 Pin Connection

Processor Power Supply Decoupling

Processor power supply decoupling is critical for reliable operation. With the 80960Hx-ready system, there are two areas of concern, each of which are described in the following subsections:

- High frequency decoupling, necessitated by the processor's high speed operation
- Low frequency decoupling, necessitated by the processor's power saving features

High Frequency Power Supply Decoupling

High frequency decoupling is critical on the Cx processor. It is especially critical on the Hx processor, because of its high speed external bus, and also because of its very fast 66 MHz internal operation.

A reliable design will include a minimum of nine 0.1 µF capacitors and nine 0.01 µF surface mount capacitors between power and ground, evenly distributed, close to the processor. The capacitors must be placed as close to the processor as possible, attached directly to the power and ground planes, or circuit board inductance will significantly reduce their effectiveness.

A typical failure mode caused by inadequate high frequency decoupling is unreliable or inconsistent program behavior. These failures are often intermittent, and are very hard to debug. Figure 6 shows a recommended layout for the high frequency capacitors, with values as shown.

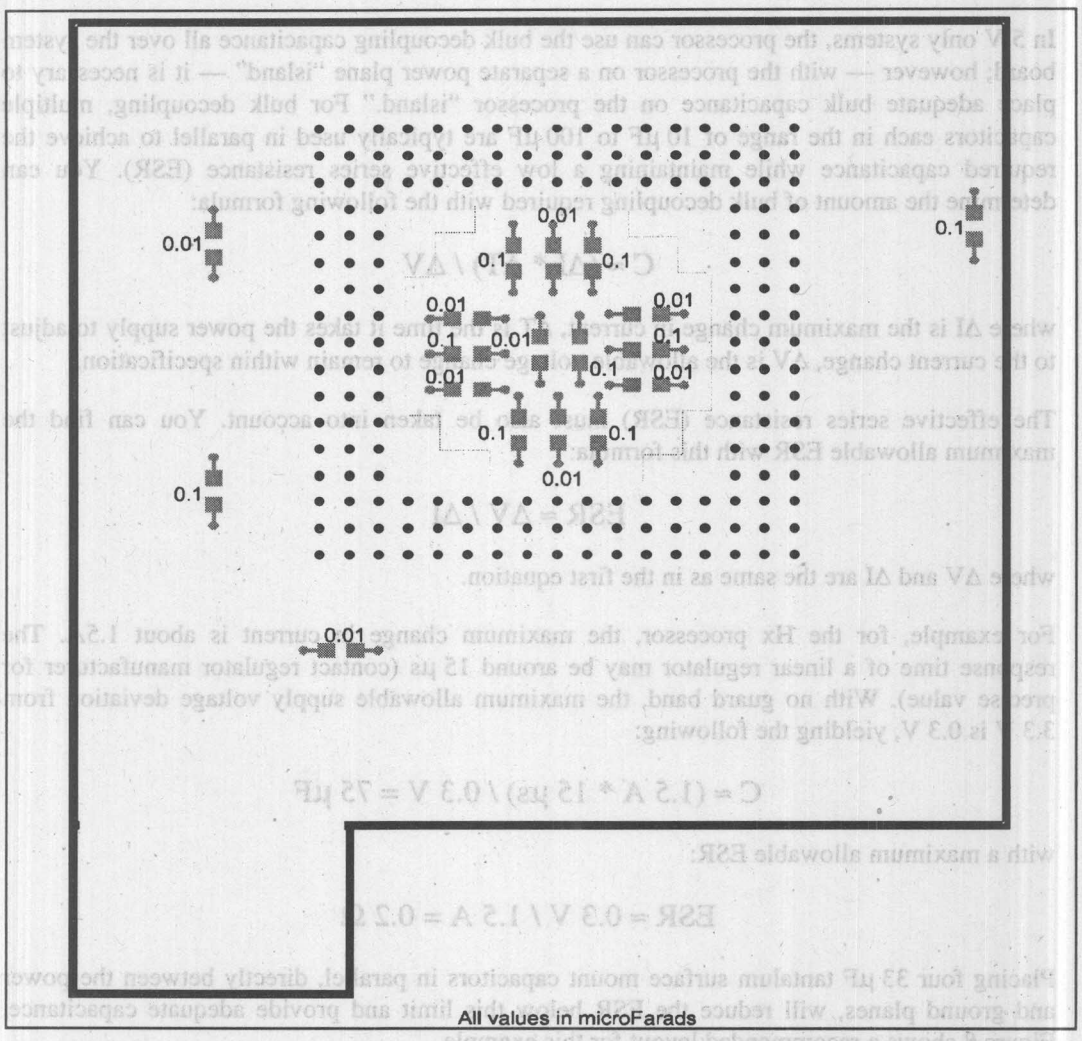


Figure 6. Recommended High-Frequency Capacitor Values and Layout

Bulk Power Supply Decoupling

Bulk, or low frequency, decoupling is needed on all i960 processors, including the Cx and Hx processors, since the Hx processor may switch between normal and low power states very quickly, causing large instantaneous current changes. To properly handle these instantaneous current changes, all designs must have adequate bulk decoupling.

In 5 V only systems, the processor can use the bulk decoupling capacitance all over the system board; however — with the processor on a separate power plane "island" — it is necessary to place adequate bulk capacitance on the processor "island." For bulk decoupling, multiple capacitors each in the range of 10 µF to 100 µF are typically used in parallel to achieve the required capacitance while maintaining a low effective series resistance (ESR). You can determine the amount of bulk decoupling required with the following formula:

$$C \approx (\Delta I * \Delta T) / \Delta V$$

where ΔI is the maximum change in current, ΔT is the time it takes the power supply to adjust to the current change, ΔV is the allowable voltage change to remain within specification.

The effective series resistance (ESR) must also be taken into account. You can find the maximum allowable ESR with this formula:

$$ESR \approx \Delta V / \Delta I$$

where ΔV and ΔI are the same as in the first equation.

For example, for the Hx processor, the maximum change in current is about 1.5A. The response time of a linear regulator may be around 15 µs (contact regulator manufacturer for precise value). With no guard band, the maximum allowable supply voltage deviation from 3.3 V is 0.3 V, yielding the following:

$$C \approx (1.5\ A * 15\ \mu s) / 0.3\ V = 75\ \mu F$$

with a maximum allowable ESR:

$$ESR \approx 0.3\ V / 1.5\ A = 0.2\ \Omega$$

Placing four 33 µF tantalum surface mount capacitors in parallel, directly between the power and ground planes, will reduce the ESR below this limit and provide adequate capacitance. Figure 8 shows a recommended layout for this example.

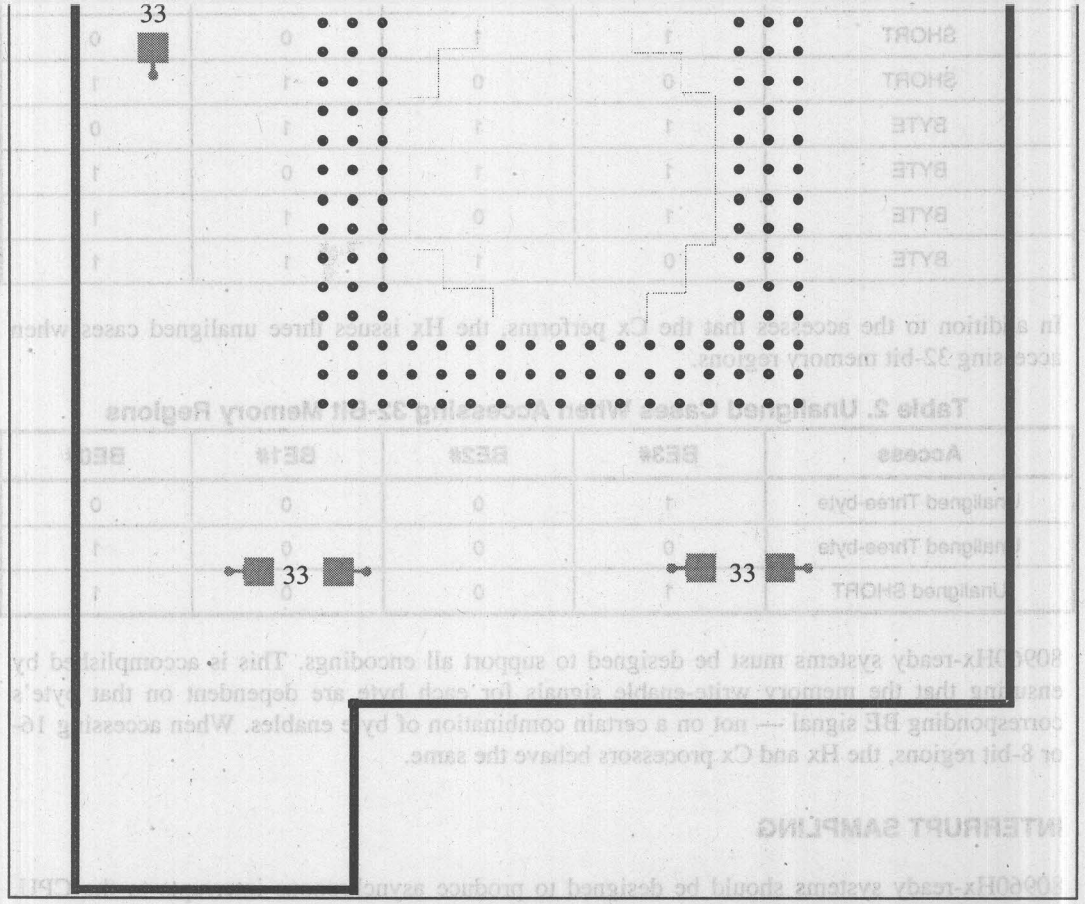


Figure 7. Recommended Bulk Decoupling Capacitor Values and Locations

BYTE ENABLE SIGNALS

The i960 Cx processors always perform aligned accesses on the bus. This means that the byte enable signals are limited to the following combinations.

Table 1. Byte Enable Signal Combinations

Access	BE3#	BE2#	BE1#	BE0#
WORD	0	0	0	0
SHORT	1	1	0	0
SHORT	0	0	1	1
BYTE	1	1	1	0
BYTE	1	1	0	1
BYTE	1	0	1	1
BYTE	0	1	1	1

In addition to the accesses that the Cx performs, the Hx issues three unaligned cases when accessing 32-bit memory regions.

Table 2. Unaligned Cases When Accessing 32-Bit Memory Regions

Access	BE3#	BE2#	BE1#	BE0#
Unaligned Three-byte	1	0	0	0
Unaligned Three-byte	0	0	0	1
Unaligned SHORT	1	0	0	1

80960Hx-ready systems must be designed to support all encodings. This is accomplished by ensuring that the memory write-enable signals for each byte are dependent on that byte's corresponding BE signal — not on a certain combination of byte enables. When accessing 16- or 8-bit regions, the Hx and Cx processors behave the same.

INTERRUPT SAMPLING

80960Hx-ready systems should be designed to produce asynchronous interrupts to the CPU. Synchronous systems such as lock-step multi-processor systems must meet input setup and hold times on the rising edges of CLKIN for the Hx, and on the falling edges of PCLK2:1 for the Cx. Interrupt pins are sampled on the rising edge of CLKIN for the Hx. Contrarily, on the Cx processors these pins are sampled on the falling edge of CLKIN. The actual sampling of the interrupt pins occurs once every two CLKIN cycles. Improper system behavior occurs if these

setup and hold times are not met in a synchronous system. An example of this is the loosing synchronous operation of multiple processors.

PARITY

A 80960Hx-ready system can implement parity when a Hx is in the CPU socket. Parity is disabled while a Cx is in the CPU socket.

Five parity pins are added to the Hx. Four of these pins, labeled DP3:0, provide byte parity for data and possess the same timing as D31:0. The fifth pin is an output labeled PCHK#. It is asserted if a parity error is detected on reads. PCHK# is asserted in the clock, following the data cycle which has incorrect parity. The Hx DP3:0 pins correspond to the CA/CF's "no connect" pins. The Hx PCHK# pin corresponds to the DACK0# pin on the CA/CF. Pull-up resistors are recommended on DP3:0. These resistors are required if parity is not being used to put the Hx parity inputs to a known state. They are also required if parity is being used when a Cx is in the system, in order to provide valid logic levels for the external parity logic. External logic will detect PCHK# high when a Cx processor is in a system. This disables external parity reporting logic.

Parity is only checked on bytes which possess a corresponding active BE signal.

CYCLE TYPE

An 80960Hx-ready system should not use cycle type pins, nor should it use DMA. The Hx uses the pins which correspond to the Cx EOP#/TC# pins for CT3:0. When ADS# is not active, the cycle type is driven to indicate whether it is executing or is in HALT mode. When ADS# is active, CT3:0 indicate the type of bus access currently being started.

Table 3. Bus Access

Cycle Type	ADS#	CT3:0
Program initiated access using 8-bit bus	0	0000
Program initiated access using 16-bit bus	0	0001
Program initiated access using 32-bit bus	0	0010
Event initiated access using 8-bit bus	0	0100
Event initiated access using 16-bit bus	0	0101
Event initiated access using 32-bit bus	0	0110
Reserved	0	0X11
Reserved	0	1XXX
Processor not halted	1	0X0X
Processor not halted	1	0X10
Reserved	1	0011
Processor in HALT mode	1	0111
Reserved for future products	1	1XXX

BSTALL

The BSTALL signal becomes active when the Hx processor can not continue execution until a pending bus transaction is completed. A load instruction followed by an instruction that uses the result of the load, causes a stall until the load is completed. A store or a load instruction, issued when the bus queues are full, also cause a stall. In this case the Hx is stalled until a bus queue entry becomes available. One of these becomes available as a result of processing a pending bus request. The instruction scheduler can cause BSTALL when the processor fetches instructions from external memory. The processor must fetch these instructions due to instruction cache misses.

The BSTALL pin can be used to provide "on demand" bus arbitration. When a system has an external bus master which is given higher priority than the Hx, it can maintain ownership of the bus until the Hx needs the bus. The Hx will assert BREQ when it has a pending bus request. When BREQ is asserted without BSTALL, the processor can continue operation even in the presence of a pending bus request. Some systems may choose to ignore this condition. Alternatively, they don't give the bus to the Hx, but instead wait until the processor is stalled. The assertion of BSTALL informs the arbitration logic of this condition.

The Cx processor does not have a BSTALL pin — the corresponding pin on the Cx is the DMA# pin. It will be driven high during normal operation (no DMA). This signals a stall condition to the external logic.

If BSTALL is used for bus arbitration in a 80960Hx-ready system, the recommendation is to logically "and" BSTALL and BREQ to indicate when the microprocessor requires the bus. By qualifying BSTALL with BREQ, the resulting signal can be used interchangeably between the Cx and Hx processors. This resulting signal is equivalent to BSTALL on a 80960Hx system, and equivalent to BREQ on a 80960Cx system.

JTAG

If boundary scan is used in an 80960Hx-ready system, a jumper should be used to connect TDI to the next device in the scan chain when a Cx is installed. The jumper should isolate the pin corresponding to TDO from the scan chain.

The Hx supports IEEE 1149.1 boundary scan. This interface consists of 5 pins: 4 input pins and 1 output pin. The JTAG interface utilizes pins used for DMA on the Cx. The Hx JTAG input pins correspond to CA/CF DREQ3:0# input pins. The JTAG output pin corresponds to a Cx DACK1# output pin.

RESERVED MEMORY

The Hx processor is not able to access external memory in the range 0xff000000 to 0xffffffff. This area is reserved for memory mapped registers. Consequently, an Hx processor cannot access the IBR of a Cx system located at 0xffffff00. The IBR of an Hx processor is located at 0xfeffff30 through 0xfeffff5f. It may be beneficial to use a single memory area mapped to two different areas.

For a system to be capable of using memory for either Hx or Cx boot up, at either 0xfexxxxxx or 0xffxxxxxx, address bit 24 should not be used in the boot area decode logic. Using this methodology, the Cx processor accesses this memory using addresses such as 0xffffff00, while the Hx processor uses addresses like 0xfeffff30.

AC TIMING

The timing of signals on the Hx differs from corresponding timing on the Cx. In general, the Hx is faster than the Cx. This generates some interesting design requirements for systems which accept either processor. Specifications for both implementations must be considered — the worst-case numbers must be used in the design.

The Hx specifications include two values for T_{ov} (output valid delay) corresponding to 5V and 3.3V memory systems. The Hx operates fastest in a 3.3V memory system, one which drives nominally 3.3V as a logic "1". The processor requires additional time to discharge a 5V "1" data signal down to a valid "0" logic level.

The worst-case sequence for AC timings is reading a "1" (high), then immediately writing a "0" (low). During the write, the processor must discharge the capacitive data bus below 1.5V to produce a valid low. It takes a few nanoseconds longer to discharge a 5V charge than a 3.3V charge.

The Hx specifications described in this section are estimates, and are subject to change when silicon becomes available.

REFERENCE CLOCK

The Cx AC timings for input and output signals are measured against the transitions of the output PCLK2:1 signals. When operating in 1x clock mode, the Cx processor input and output clocks are synchronized. T_{CP}, the CLKIN to PCLK2:1 delay, is +/- 2ns in 1x mode. When operating in 2x mode, the output clock edges are delayed from the input clocks. In 2x mode, T_{CP} is 2 to 25ns at 33 MHz.

The Hx has no output clocks; Hx AC timings are specified according to the input clock.

One of two clocking methods are recommended for a 80960Hx-ready system:

- External logic can be clocked with PCLK2:1 when a Cx is plugged into the socket. It can be clocked with CLKIN when using a Hx processor.
- Always use CLKIN to clock external logic for either a Cx or Hx processor.

For the first of the above recommendations, a method of clock selection must be implemented. Jumpers can be used to select either CLKIN or PCLK2:1 (to route to the synchronous logic within the system). This is a simple methodology because the clocked logic performs the same function with either processor. One benefit derived from this is that the clocks used by external logic are always the processor's reference clocks.

CLKIN is the reference for the Hx; for the Cx it may be away from the reference (PCLK2:1) by as much as 2ns in 1x clock mode, or 25ns in 2x clock mode. This offset must be considered when analyzing system timing. Due to the wide range of possible delays, it is not practical to use 2x clock mode when using CLKIN for the external logic. The Hx does not support a 2x clock input.

INPUT/OUTPUT TIMING

Input pins specify setup and hold times according to the processor reference clock.

Input signals must be stable between the minimum input setup and hold times. This is the time when the signals are being latched internally within the processor. For minimum input setup and hold values, use the figures with the largest **maximum** values between the two devices.

AC timing parameters for output signals include both a minimum output hold time and a maximum output valid delay. The minimum output hold time specifies the time after a clock during which a signal continues to be valid from the previous state. The maximum output valid delay specifies the maximum time necessary for a signal to switch states.

Output signals switch between the minimum output hold and the maximum output valid times. For minimum output hold, the smallest **minimum** value of the different devices should be used. Maximum output valid delay is the largest **maximum** value of the different devices.

The combined specification for AC timings differs from the sole specification of either a Cx or Hx processor. An application which accepts either a Cx or Hx must operate over this wider range of timing. For example, pins D31:0 are bi-directional and require both input and output timing analysis. The AC timings used in this example are subject to change; refer to current data sheet for actual values.

		D0 (80960CF)	D0 (80960Hx)	Combined
Output timing	TOH (min)	3 ns	1.5 ns	1.5 ns
5V I/O	TOV (max)	16 ns	13 ns	16 ns
3.3V I/O	TOV (max)	na	10 ns	16 ns
Input timing	TIS (min)	3 ns	6 ns	6 ns
	TIH (min)	5	1.5	5

During a write cycle:

- An 80960CF outputs data within a 13 ns window — between 3 and 16 ns after the corresponding clock edge.
- For a 5V I/O system, an 80960 Hx outputs data within a 11.5 ns window — between 1.5 and 13 ns.
- For a 3.3V I/O system, an 80960 Hx outputs data within an 8.5 ns window — between 1.5 and 10 ns.

The combination of these specifications leads to a 14.5 ns window — between 1.5 and 16 ns. Minimum output hold (T_{OH}) analysis must be performed using 1.5 ns. This is the worst case time. Maximum output delay (T_{OV}) analysis must be performed using 16 ns, worst case. Similar "widening" of specifications also occur on input timings.

PINOUT

The following table highlights the differences between the Hx and Cx processors. Differences are indicated with a heavier line around the table cell. Table 4 shows the pin differences between the Cx and Hx. The "comments" section describes recommended usage in an 80960Hx-ready system.

PGA Pin	Hx Signal Name	Cx Signal Name	PGA Pin	Hx Signal Name	Cx Signal Name	PGA Pin	Hx Signal Name	Cx Signal Name	PGA Pin	Hx Signal Name	Cx Signal Name
A1	VSS	NC	C9	VSS	VSS	J15	VSS	VSS	Q10	VSS	VSS
A2	FAIL#	FAIL#	C10	VSS	VSS	J16	VCC	VCC	Q11	VSS	VSS
A3	DP0	NC	C11	VSS	VSS	J17	A10	A10	Q12	SUP#	SUP#
A4	DP2	NC	C12	VSS	VSS	K1	D13	D13	Q13	A30	A30
A5	VOLDET	NC	C13	CLKIN	CLKIN	K2	VCC	VCC	Q14	A28	A28
A6	TRST#	DREQ1#	C14	VCC	CLKMODE	K3	VSS	VSS	Q15	A24	A24
A7	TDI	DREQ3#	C15	XINT4#	XINT4#	K15	VSS	VSS	Q16	A21	A21
A8	TDO	DACK1#	C16	XINT6#	XINT6#	K16	VCC	VCC	Q17	A18	A18
A9	NC	DACK2#	C17	XINT7#	XINT7#	K17	A11	A11	R1	D24	D24
A10	NC	DACK3#	D1	D5	D5	L1	D15	D15	R2	D27	D27
A11	CT0	EOP/TC0#	D2	D2	D2	L2	D14	D14	R3	D31	D31
A12	CT1	EOP/TC1#	D3	NC	NC	L3	VSS	VSS	R4	BTERM#	BTERM#
A13	CT2	EOP/TC2#	D15	NMI#	NMI#	L15	VSS	VSS	R5	HOLD	HOLD
A14	CT3	EOP/TC3#	D16	A2	A2	L16	A13	A13	R6	ADS#	ADS#
A15	XINT1#	XINT1#	D17	A3	A3	L17	A12	A12	R7	VCC	VCC
A16	RESET#	RESET#	E1	D7	D7	M1	D16	D16	R8	VCC	VCC
A17	XINT2#	XINT2#	E2	D4	D4	M2	VCC	VCC	R9	BE0#	BE0#
B1	BOFF#	BOFF#	E3	D0	D0	M3	VSS	VSS	R10	VCC	VCC
B2	STEST	STEST	E15	VCC	VCC	M15	VSS	VSS	R11	VCC	VCC
B3	DP1	NC	E16	A4	A4	M16	VCC	VCC	R12	BSTALL	DMA#
B4	DP3	NC	E17	A5	A5	M17	A14	A14	R13	BREQ	BREQ
B5	TCK	DREQ0#	F1	D8	D8	N1	D17	D17	R14	A29	A29
B6	TMS	DREQ2#	F2	D6	D6	N2	D18	D18	R15	A26	A26
B7	VCC	VCC	F3	VCC	VCC	N3	VCC	VCC	R16	A23	A23
B8	PCHK#	DACK0#	F15	VSS	VSS	N15	VCC	VCC	R17	A22	A22
B9	VCC	VCC	F16	VCC	VCC	N16	A16	A16	S1	D25	D25
B10	VCCPLL	VCCPLL	F17	A6	A6	N17	A15	A15	S2	D29	D29
B11	VCC	VCC	G1	D9	D9	P1	D19	D19	S3	READY#	READY#
B12	VCC	VCC	G2	VCC	VCC	P2	D20	D20	S4	HOLDA	HOLDA
B13	NC	PCLK2	G3	VSS	VSS	P3	D22	D22	S5	BE3#	BE3#
B14	NC	PCLK1	G15	VSS	VSS	P15	A20	A20	S6	BE2#	BE2#
B15	XINT0#	XINT0#	G16	A7	A7	P16	A19	A19	S7	BE1#	BE1#
B16	XINT3#	XINT3#	G17	A8	A8	P17	A17	A17	S8	BLAST#	BLAST#
B17	XINT5#	XINT5#	H1	D11	D11	Q1	D21	D21	S9	DEN#	DEN#
C1	D3	D3	H2	D10	D10	Q2	D23	D23	S10	W/R#	W/R#
C2	D1	D1	H3	VSS	VSS	Q3	D26	D26	S11	DT/R#	DT/R#
C3	ONCE#	ONCE#	H15	VSS	VSS	Q4	D28	D28	S12	WAIT#	WAIT#
C4	VSS	NC	H16	VCC	VCC	Q5	D30	D30	S13	D/C#	D/C#
C5	VCC5	NC	H17	A9	A9	Q6	VCC	VCC	S14	LOCK#	LOCK#
C6	VCC	VCC	J1	D12	D12	Q7	VSS	VSS	S15	A31	A31
C7	VSS	VSS	J2	VCC	VCC	Q8	VSS	VSS	S16	A27	A27
C8	VSS	VSS	J3	VSS	VSS	Q9	VSS	VSS	S17	A25	A25

Table 4. 80960Cx/80960Hx Pin Differences

			- CA/CF. VSS - Hx
A6	DREQ1#	TRST#	When active (low), causes TAP controller (IEEE 1149.1) to go to Test_Logic_Reset state. This pin should be connected to RESET# through a 10KΩ resisitor.
A7	DREQ3#	TDI	OK to pull-up or drive. If it is driven low, be sure DMA is disabled. If JTAG is used with a Cx in the system, this signal should be connected to TDI of next device in the chain, via a jumper.
A8	DACK1#	TDO	For Cx, this pin will always be high when DMA is not in use. Because this pin is an output, use a jumper or external logic to disconnect this pin when using the Cx.
A9	DACK2#	NC	No connection
A10	DACK3#	NC	No connection
A11-A14	EOP/TC#3:0	CT3:0	Use pull-ups. An output of 1111 indicates a Cx processor is in the system. Indicates the cycle type if ADS# is active. Indicates when the processor is halted if ADS# is not active.
B3	NC	DP1	Should be pulled up with a resistor. In a system with parity, connect to the parity bit which corresponds to D15:8.
B4	NC	DP3	Should be pulled up with a resistor. In a system with parity, connect to the parity bit which corresponds to D31:28.
B5	DREQ0#	TCK	Connect to Test Clock of 1149.1 interface. This pin should be pulled high when not in use.
B6	DREQ2#	TMS	Connect to Test Mode Select of 1149.1 interface. This pin should be pulled high when not in use.
B8	DACK0#	PCHK#	Connect to external parity error recovery/reporting logic. Cx will not generate or check parity.
B13	PCLK2	NC	OK to drive Hx with CLKIN for compatibility.

DESIGN GUIDELINE SUMMARY

A system can be designed which accepts either a Hx or Cx processor. The following items summarize the guidelines discussed in this paper:

- Don't use the DMA controller on the Cx processor
- Isolate V_{cc} for the CPU. Hx = 3.3V; Cx = 5V
- Provide 5V reference voltage for Hx (VCC5)
- Use CLKIN for system timing
- Combine AC specifications for timing analysis
- Accommodate new BE3:0# encodings
- Use pull-up resistors on parity signals
- Connect additional V_{ss} signals
- If using JTAG boundary scan, bypass Cx in the JTAG chain

S1+	PCLK1	NC	OK to drive Hx with CLKIN for compatibility.
C+	NC	V_{SS}	Connect to V_{SS}.
C5	NC	VCC5	Connect to 5V through a 100 Ohm resistor if inputs can be driven from 5V logic. Connect directly to 3.3V if inputs are not driven by 5V logic.
C1+	CLKMODE	VCC	Connect to processor's V_{CC}.
R12	DMA#	BSTALL	Can use for arbitration.

DESIGN GUIDELINE SUMMARY

A system can be designed which accepts either a Hx or Cx processor. The following items summarize the guidelines discussed in this paper:

- Don't use the DMA controller on the Cx processor
- Isolate V_{CC} for the CPU. Hx = 3.3V; Cx = 5V
- Provide 5V reference voltage for Hx (VCC5)
- Use CLKIN for system timing
- Combine AC specifications for timing analysis
- Accommodate new BE3:0# encodings
- Use pull-up resistors on parity signals
- Connect additional V_{SS} signals
- If using JTAG boundary scan, bypass Cx in the JTAG chain

PRODUCT PREVIEW

intel

AP-703

APPLICATION NOTE

DRAM Controller for 33 MHz i960® CA/CF Microprocessors

February 2, 1995

Order Number: 272627-001

REVISION HISTORY

Changes from Rev 001 to Rev 002 include:

Section	Description
AC TIMING	Two paragraphs added after the first: The Hx specifications include two values for T_{ov} (output valid delay) corresponding to 5V and 3.3V memory systems. The Hx operates fastest in a 3.3V memory system, one which drives nominally 3.3V as a logic "1". The processor requires additional time to discharge a 5V "1" data signal down to a valid "0" logic level. The worst-case sequence for AC timings is reading a "1" (high), then immediately writing a "0" (low). During the write, the processor must discharge the capacitive data bus below 1.5V to produce a valid low. It takes a few nanoseconds longer to discharge a 5V charge than a 3.3V charge.
INPUT/OUTPUT TIMING	The second bulleted item in this section changed. WAS: • A Hx outputs data within a 7 ns window — between 1.5 and 8.5 ns. IS: • For a 5V I/O system, an 80960 Hx outputs data within a 11.5 ns window — between 1.5 and 13 ns. • For a 3.3V I/O system, an 80960 Hx outputs data within an 8.5 ns window — between 1.5 and 10 ns.
INPUT/OUTPUT TIMING	Numbers changed in the table that compares Input/Output timing of the Hx and Cx. Output Timing TOV (max) for the Hx (3.3V) changed from 8.5ns to 10ns. Output Time TOV (max) for the Hx (5V) was added (13ns). Input timing for TIS(min) for the Hx changed from 5ns to 6ns. Input timing for TIS(min) for the "Combined" changed from 5ns to 6ns.
Table 4. 80960Cx/80960Hx Pin Differences	Pin definition for A6, last sentence, changed. WAS: This pin should be pulled high when not in use. IS: This pin should be connected to RESET# through a 10KΩ resisitor.

1.0 INTRODUCTION

This application note describes a DRAM controller for use with the i960® CA/CF 33 MHz microprocessors. Other application notes are available which describe DRAM controllers for the i960 CF and Jx processors; see Section 7.0, RELATED INFORMATION for ordering information.

This DRAM controller's design features include:

- Interleaved design
- Can use standard 70 ns DRAM SIMM
- 3-0-0-0/2-0-0-0 back-to-back/idle bus wait state burst reads at speeds up to 33 MHz
- 3-1-1-1/2-1-1-1 back-to-back/idle bus wait state burst writes at speeds up to 33 MHz
- No delay lines

This application note contains some general DRAM controller theory as well as this design's state machine definitions and timing diagrams. It also contains the PLD equations which were used to build and test the prototype design. Timing analysis was verified with Timing Designer*. PLD equations were created in ABEL* as a device-independent design. Schematics were developed with OrCAD*. The timing analysis, schematics and PLD files are available through Intel's America's Application Support BBS, at (916) 356-3600.

2.0 OVERVIEW

This section provides an overview of DRAM SIMM operation and the concept of memory interleaving. It also describes the i960 Cx microprocessor's burst capabilities.

2.1 Page Mode DRAM SIMM Review

Page mode DRAM allows faster memory access by keeping the same row address while selecting random column addresses within that row. A new column address is selected by deasserting $\overline{CAS}$ while keeping $\overline{RAS}$ active and then asserting $\overline{CAS}$ with the new column address valid to the DRAM. Page mode operation works very well with burst buses, such as those in the i960 CA/CF processors, in which a single address cycle can be followed by multiple data cycles.

All $\overline{WE}$ pins on each SIMM are tied to a common $\overline{WE}$ line; this feature requires the use of early write cycles. In an early write cycle, write data is referenced to the falling edge of $\overline{CAS}$, not the falling edge of $\overline{WE}$.

Each SIMM also has four $\overline{CAS}$ lines, one for every eight (nine) bits in a 32-bit (36-bit) SIMM module. The four $\overline{CAS}$ lines control the writing to individual bytes within each SIMM.

2.2 Bank Interleaving

Interleaving significantly improves memory system performance by overlapping accesses to consecutive addresses. Two-way interleaving is accomplished by dividing the memory into two 32-bit banks (also referred to as "leaves"):

- one bank for even word addresses (A2=0)
- one bank for odd word addresses (A2=1)

The two banks are read in parallel and the data from the two banks is multiplexed onto the processor's data bus. This overlaps the wait states of:

- the second access with the first
- the third access with the second
- the fourth access with the third

Figure 1 shows DRAM with a 2-1-1-1 quad word burst read wait state profile being interleaved to generate a 2-0-0-0 wait state system.

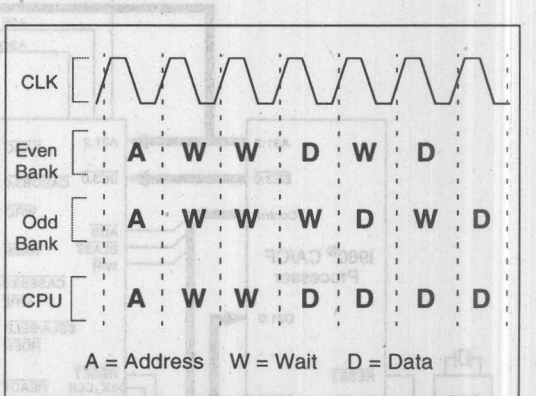


Figure 1. Two-Way Interleaving

2.3 Burst Capabilities for 32-Bit Bus

A bus access starts by asserting $\overline{ADS}$ in the address cycle, and ends by asserting $\overline{BLAST}$ in the last data cycle. Figure 2 shows $\overline{ADS}$ and $\overline{BLAST}$ timings for a quad-word access.

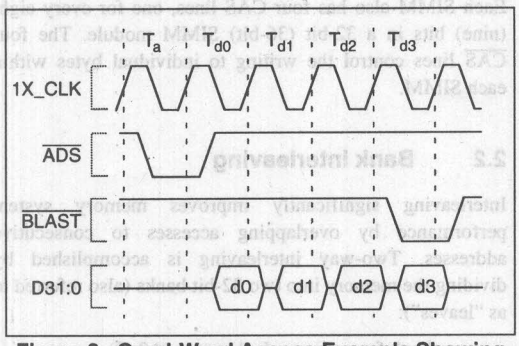


Figure 2. Quad-Word Access Example Showing ADS and BLAST Timings

The i960 Cx processor's burst protocol requires:

- Quad-word and triple-word requests always start on quad word boundaries (A3 = 0, A2 = 0).
- Double-word requests always start on double word boundaries (A3 = X, A2 = 0).
- Single-word requests can start on any word boundary (A3 = X, A2 = X).
- Any request starting on an odd word boundary never bursts (A3 = X, A2 = 1).

3.0 BASIC DRAM CONTROLLER

The DRAM controller comprises four distinct blocks: control logic, address path, data path, and the DRAM SIMMS. This section describes each block.

3.1 Control Logic

The DRAM controller is centered around a four-bit state machine which controls DRAM bank accesses and DRAM refresh. All timings are generated based on the four-bit state machine's outputs. Some states are used for both read and write accesses. The state machine uses the W_R signal from the processor to distinguish between reads and writes. This technique allows the state machine to use fewer states; therefore, fewer output bits.

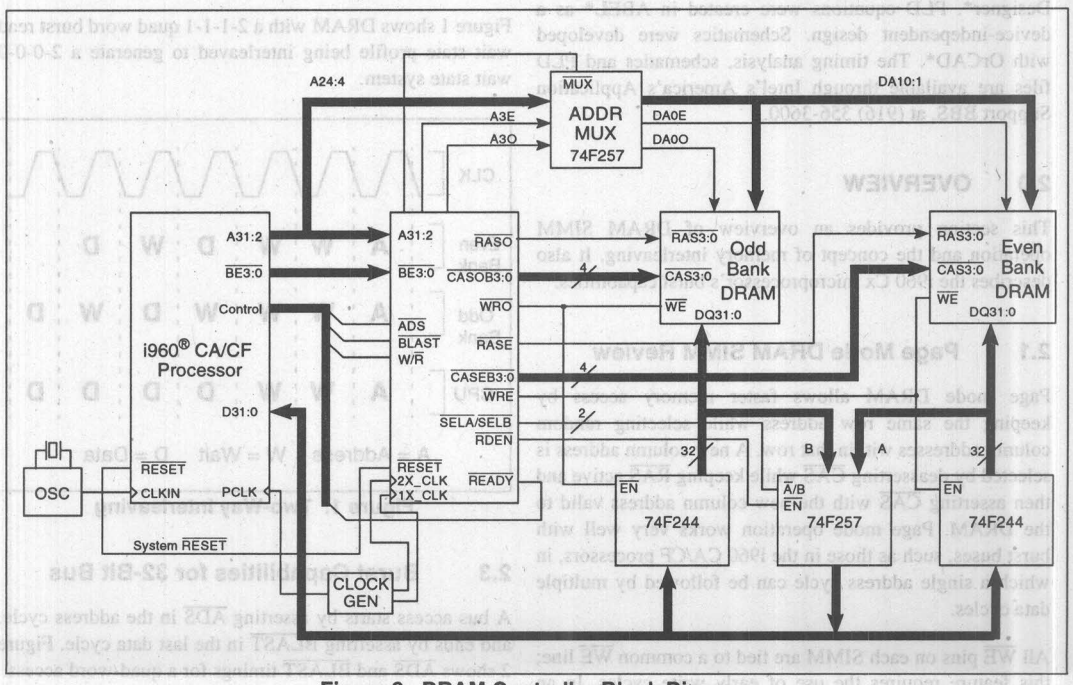


Figure 3. DRAM Controller Block Diagram

3.1.1 Refresh Logic (CAS-before-RAS)

Typically DRAM needs to be refreshed every 15.6 µs. In this design, due to power requirements needed to refresh an entire DRAM array, one bank is refreshed at a time. The DRAM controller uses an eight-bit counter to generate refresh requests. A refresh request is generated every 7.8 µs. The DRAM controller toggles between refreshing each bank every 7.8 µs which means each bank is effectively refreshed every 15.6 µs.

A refresh request has priority over a processor request. When a processor and a refresh request occur simultaneously, the DRAM controller sequences a refresh to the appropriate DRAM bank while the PENDING state machine posts the processor request. The pending request is then serviced after the refresh is completed.

An eight-bit synchronous down counter is used to generate refresh requests. The counter is clocked using 1X_CLK clock. $\overline{\text{REFREQ}}$ is asserted each time the counter reaches zero. Counting is inhibited when the counter reaches zero. The counter is reloaded with 0xff and counting resumes after the ACCESS state machine services the refresh. During reset, the counter is loaded with 0xff.

3.1.2 Clock Generation

In the tested design, Motorola* MC88915 low skew CMOS PLL generates the clock signals for the DRAM controller. The MC88915 uses PCLK2 as an input, and produces four very low skew copies of PCLK2, as well as a 2x PCLK. At 33 MHz, the maximum skew between PCLK2 and any of the MC88915 outputs was calculated to be ±1 ns, while the skew between any of the individual outputs is ±750 ps under equal loading conditions. All clock lines are terminated with 22 ohm series resistors.

3.1.3 Wait State Profile

The DRAM Controller uses the processor's $\overline{\text{READY}}$ signal to control wait states. The MCON register is initialized as follows: ($N_{XAD} = N_{XDD} = N_{XDA} = 0$). Table 1, Wait State Profiles (33 MHz), provides the wait state profiles for read and write accesses up to 33 MHz. Back-to-back accesses require an extra wait state to meet RAS precharge time. Therefore, to meet the RAS precharge time required, the first data access uses three wait state cycles as opposed to two wait state cycles for idle bus DRAM accesses.

Table 1. Wait State Profiles (33 MHz)

Access Type	Wait State Profile	
	Back-To-Back	*Idle Bus*
Quad Word Read	3-0-0-0	2-0-0-0
Triple Word Read	3-0-0	2-0-0
Double Word Read	3-0	2-0
Single Word Read	3	2
Quad Word Write	3-1-1-1	2-1-1-1
Triple Word Write	3-1-1	2-1-1
Double Word Write	3-1	2-1
Single Word Write	3	2

3.2 Address Path

Figure 4 is a block diagram of the address path logic. The 2-to-1 multiplexers combine the row and column addresses into a singular row/column address that the DRAM requires. DA0E and DA0O equivalent signals are generated, one for each bank. DA0E and DA0O are generated by using A3E and A3O respectively. DA0E and DA0O are the only address bits that increment during bursts. The timing of these signals during bursts is critical for proper operation.

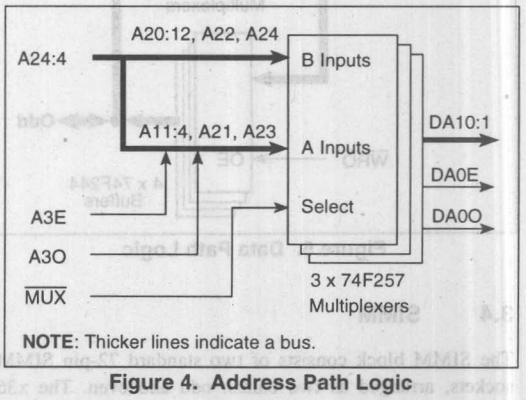


Figure 4. Address Path Logic

3.3 Data Path

As shown in Figure 5, Data Path Logic, there is one data path for reads and a separate data path for writes. The read path uses 74F257 2:1 multiplexers to prevent contention between the two DRAM banks. $\overline{\text{CAS}}$ can be active for both

banks at the same time, necessitating use of the multiplexers. The multiplexer outputs are enabled only during reads by the $\overline{RDEN}$ signal. The multiplexers are switched using $\overline{SELA}$ and $\overline{SELB}$. These signals are derived from the states of the ACCESS state machine and address A2.

The write data path consists of eight 8-bit 74F244 buffers, four for each bank. The buffer outputs are enabled by $\overline{WRE}$ and $\overline{WRO}$.

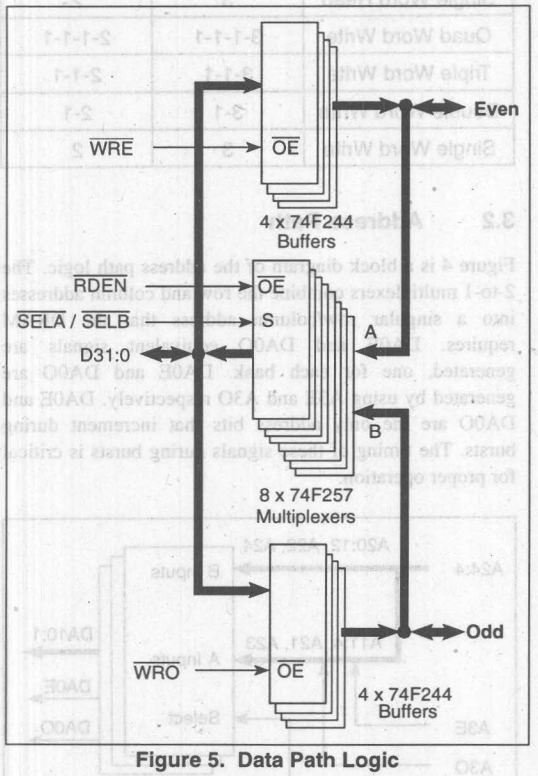


Figure 5. Data Path Logic

3.4 SIMM

The SIMM block consists of two standard 72-pin SIMM sockets, arranged as two banks: odd and even. The x36 SIMM parity bits are not used in this design. However, x36 SIMMs are standard for PCs and workstations and are readily available. The only penalty is more loading on the address and control lines due to the extra DRAM devices of x36 SIMM. In the tested design, all address and control lines to the SIMMs are terminated with 22 ohm resistors.

4.0 STATE MACHINES AND SIGNALS

This section describes the state machines and signals used in this design. Most of the state machines are simple and the PLD equations can be referenced in APPENDIX A. The ACCESS state machine is the most complex of all the state machines; for that reason, this application note provides more detail on the operations of this state machine. In this design, some state machines are clocked with the 1X_CLK clock (bus clock frequency) and others with the 2X_CLK clock (twice the bus clock frequency).

All PLD equations are written in ABEL. APPENDIX A, PLD EQUATIONS contains a listing of the PLD equation file. State machine transitions described here follow the ABEL conventions for logic operators:

- ! represents NOT, bit-wise negation
- & represents AND
- # represents OR

To follow ABEL conventions, active LOW signals (such as ADS) already have a polarity assigned. For example, in the state machines, ADS refers to the asserted state (LOW) and !ADS refers to the non-asserted state (HIGH).

4.1 ACCESS State Machine

The ACCESS state machine, the "heart" of the DRAM controller, is implemented as a four-bit state machine. See Figure 6, Basic ACCESS State Machine. It is responsible for sequencing accesses and refreshes to the DRAM banks.

From the IDLE state, the access state machine is sequenced based on these three events:

- Refresh requests from the counter
- DRAM requests from the processor
- PENDING state machine requests

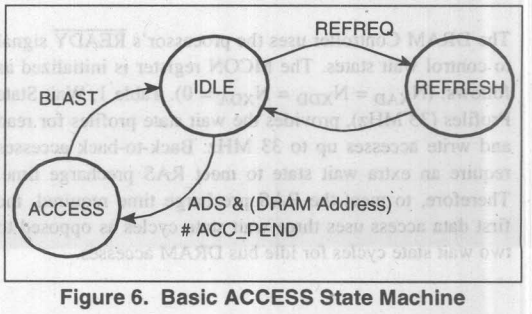


Figure 6. Basic ACCESS State Machine

4.2 PENDING State Machine

The PENDING state machine is a one-bit state machine which monitors DRAM requests from the processor. This is necessary because a DRAM refresh has priority over a processor request. Therefore, this state machine is used to post the processor request. The state machine gets reset once the ACCESS state machine starts sequencing the pending request. The state machine generates $\overline{\text{ACC\_PEND}}$.

4.3 ODDACCESS State Machine

The ODDACCESS state machine is clocked using the 1X_CLK clock. It is a one-bit state machine which monitors the initial state of the processor's address A2. Several state machines in this design use the output of this state machine as inputs. Address A2 from the processor indicates whether an access starts on an even or odd word boundary. The ACCESS state machine uses this bit extensively. It is important to latch address A2 because the processor toggles address A2 on burst accesses. This state machine generates $\overline{\text{LA2}}$.

4.4 BANKSELA State Machine

The BANKSELA state machine is a one-bit state machine which is used to control the data multiplexer, primarily to select between even or odd data during read accesses. This state machine is clocked using the 1X_CLK clock. It generates $\overline{\text{SELA}}$.

4.5 BANKSELB State Machine

BANKSELB is a one-bit state machine which controls the data multiplexer, primarily to select between even or odd data during read accesses. This state machine is clocked using the 1X_CLK clock. It generates $\overline{\text{SELB}}$.

4.6 ADDRMUX State Machine

The ADDRMUX state machine is a one-bit state machine which is used to control the address multiplexers, essentially to select between row or column addresses. It is clocked using the 2X_CLK clock. This state machine generates $\overline{\text{MUX}}$. This signal is a delayed version of $\overline{\text{RASE}}$. Delaying the switching of the row address by one 2X_CLK clock cycle provides ample row address hold time (t_{RAH}) required by the DRAM. The row address is selected while $\overline{\text{MUX}}$ is high; otherwise, the column address is selected.

4.7 A3EVEN State Machine

The A3EVEN state machine is a one-bit state machine which is toggled on burst accesses to select the next data word (next column data). The state machine is initially loaded with the value of the processor's address A3 and then toggled for the next data access. This state machine is clocked using the 2X_CLK clock, and generates A3E. This signal is an input to the address multiplexer.

4.8 A3ODD State Machine

The A3ODD state machine is a one-bit state machine and has the same functionality as the A3EVEN state machine. This state machine generates A3O.

4.9 RFEVENBK State Machine

The RFEVENBK state machine is a one-bit state machine which is used to indicate which of the two banks (even or odd) to refresh. The two banks are refreshed separately. The even bank is refreshed when the RFEVENBK state machine is active; otherwise, the odd bank is refreshed. The output of this state machine is toggled on every refresh. This state machine generates $\overline{\text{REFEVEN}}$.

4.10 CASPIPE State Machine

The CASPIPE state machine is a one-bit state machine which generates a pipelined $\overline{\text{CAS}}$ signal one 2X_CLK clock cycle earlier. The output of this state machine is then fed to the CASE_B3:0 state machines where it is reconstructed to drive the $\overline{\text{CAS}}$ lines of the even bank. This state machine generates $\overline{\text{CASEE}}$.

4.11 CASPIPO State Machine

CASPIPO is a one-bit state machine which generates a pipelined $\overline{\text{CAS}}$ signal one 2X_CLK clock cycle earlier. Its output is then fed to the CASO_B3:0 state machines where it is reconstructed to drive the $\overline{\text{CAS}}$ lines of the odd bank. This state machine generates $\overline{\text{CASOO}}$.

4.12 CASE_B3:0 State Machines

The CASE_B3:0 state machines control the $\overline{\text{CAS}}$ pins of the even bank. CASE_B0 controls the least significant byte and CASE_B3 controls the most significant byte. The CASE_B0 state machine generates $\overline{\text{CASEB0}}$, and the CASE_B3 state machine generates $\overline{\text{CASEB3}}$. $\overline{\text{CASEB0}}$ is

asserted when $\overline{\text{CASEE}}$ and the processor's $\overline{\text{BE0}}$ signal are asserted. $\overline{\text{CASEB3}}$ is asserted when $\overline{\text{CASEE}}$ and the processor's $\overline{\text{BE3}}$ signal are asserted. The CASE_B3:0 state machines are clocked using the 2X_CLK clock.

4.13 CASO_B3:0 State Machines

The CASO_B3:0 state machines control the $\overline{\text{CAS}}$ pins of the odd bank. CASO_B0 controls the least significant byte and CASO_B3 controls the most significant byte. The CASO_B0 state machine generates $\overline{\text{CASOB0}}$, and the CASO_B3 state machine generates $\overline{\text{CASOB3}}$. $\overline{\text{CASOB0}}$ is asserted when $\overline{\text{CASOO}}$ and the processor's $\overline{\text{BE0}}$ signal are asserted. $\overline{\text{CASOB3}}$ is asserted when $\overline{\text{CASOO}}$ and the processor's $\overline{\text{BE3}}$ signal are asserted. The CASO_B3:0 state machines are clocked using the 2X_CLK clock.

4.14 RASEVEN State Machine

RASEVEN is a one-bit state machine which is used to generate the $\overline{\text{RAS}}$ signals for the even bank. It is clocked using the 2X_CLK clock; it generates $\overline{\text{RASE}}$.

4.15 RASODD State Machine

RASODD is a one-bit state machine which is used to generate the $\overline{\text{RAS}}$ signals for the odd bank. It is clocked using the 2X_CLK clock; it generates $\overline{\text{RASO}}$.

4.16 $\overline{\text{SRASE}}$ State Machine

SRASE is a one-bit state machine which is used to monitor back-to-back DRAM accesses. It is generated by shifting $\overline{\text{RASE}}$ by one 1X_CLK clock cycle. This state machine generates $\overline{\text{SRASE}}$. By using this signal's state, the DRAM controller can eliminate one wait state cycle for accessing the first data word. Back-to-back accesses require an extra wait state cycle to satisfy the RAS precharge time (t_{RP}).

4.17 $\overline{\text{RDEN}}$ Signal

$\overline{\text{RDEN}}$ is asserted while a DRAM read is in progress. It controls the data multiplexers output enables.

4.18 $\overline{\text{WRE}}$ Signal

$\overline{\text{WRE}}$ is asserted while a DRAM write is in progress. It controls the even leaf $\overline{\text{WE}}$ lines to perform early writes. It also controls the even data path buffers output enables.

4.19 $\overline{\text{WRO}}$ Signal

$\overline{\text{WRO}}$ is asserted while a DRAM write is in progress. It controls the odd leaf $\overline{\text{WE}}$ lines to perform early writes. It also controls the odd data path buffers output enables.

4.20 $\overline{\text{REFREQ}}$ Signal

$\overline{\text{REFREQ}}$, an active low signal, is the output of an eight-bit counter. The counter is clocked using 1X_CLK. $\overline{\text{REFREQ}}$ is asserted when the counter reaches zero. The ACCESS state machine uses $\overline{\text{REFREQ}}$ to sequence refreshes.

5.0 DRAM CONTROLLER ACCESS FLOW

This section explains how the ACCESS state machine is sequenced while reading, writing, and refreshing DRAM. Examples used are:

- quad-word read
- single-word read
- quad-word write
- single-word write
- refresh

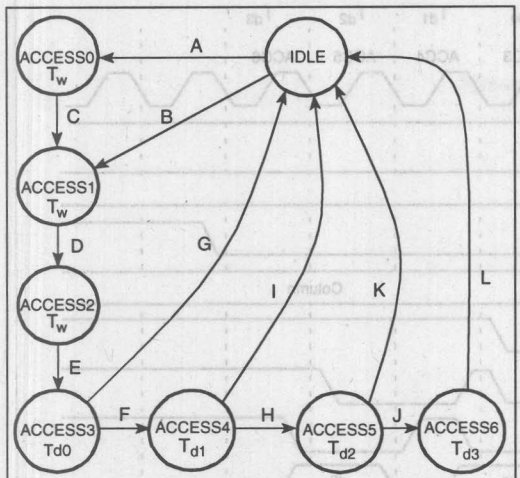
The examples in this application note assume back-to-back DRAM accesses or pending accesses. For example, the first data access of a DRAM request uses three wait states for both reads and writes. For idle bus accesses, the ACCESS0 state is skipped, allowing only two wait states.

Refer to APPENDIX A, PLD EQUATIONS. The ACCESS state machine uses $\overline{\text{SRASE}}$ to detect back-to-back accesses.

$\overline{\text{RDEN}}$ is asserted during read accesses while $\overline{\text{WRE}}$ and $\overline{\text{WRO}}$ are asserted during write accesses.

5.1 Quad-Word Read

Figure 7 shows the state diagram for a quad-word read state diagram. Figure 8 shows a quad-word read timing diagram. This state diagram shows the paths for triple-, double-, and single-word reads. Single-word reads which are aligned on odd word boundaries use a different path; therefore, a separate example is used to explain that state machine path.




```
A =  ADS & !REFREQ & !ACC_PEND & DRAMADDR & SRASE
   # !REFREQ & ACC_PEND
B =  ADS & !REFREQ & !ACC_PEND & DRAMADDR
     & !SRASE & !A2
C =  LA2
D =  UNCONDITIONAL
E =  UNCONDITIONAL
F =  !BLAST
G =  BLAST & LA2
H =  !BLAST
I =  W_R & BLAST
J =  !BLAST
K =  BLAST
L =  UNCONDITIONAL
```

Figure 7. Quad-Word Read State Diagram

From the IDLE state, the machine enters the ACCESS0 state due to a processor request or a pending processor request. At the end of the IDLE state, the A3EVEN and A3ODD state machines are loaded with the processor's address A3 and the ODDACCESS state machine is loaded with the processor's address A2. While in the IDLE state, $\overline{MUX}$ is deasserted, which selects the row address.

At the end of the ACCESS0 state, $\overline{RASE}$ and $\overline{RASO}$ are asserted. The machine then proceeds to ACCESS1 state.

In the middle of the ACCESS1 state, $\overline{MUX}$ is asserted. This causes the column address to be selected. At the end of ACCESS1, $\overline{CASEE}$ is asserted. From ACCESS1, the machine enters ACCESS2 state.

In the middle of the ACCESS2 state, $\overline{CASEB}$3:0 are asserted if $\overline{CASEE}$ and the respective byte enable signals from the processor are asserted. At the end of ACCESS2, $\overline{CASOO}$ is asserted or $\overline{BLAST}$ is not asserted. The machine then proceeds to the ACCESS3 state.

The ACCESS3 state is the first data cycle (T_{d0}) for read requests which are aligned on even word boundaries (A2=0). In the middle of the ACCESS3 state, $\overline{CASOB}$3:0 are asserted if $\overline{CASOO}$ and the respective byte enable signals from the processor are asserted, whereas $\overline{CASEE}$ is deasserted. At the end of ACCESS3, $\overline{CASEB}$3:0 are deasserted if they were earlier asserted. $\overline{CASEB}$3:0 are deasserted because $\overline{CASEE}$ is sampled deasserted. $\overline{CASEE}$ is reasserted at the end of ACCESS3 to initiate the third data (T_{d2}) access. The pending state machine is reset before leaving this state. From ACCESS3, the machine can proceed to either the IDLE state or the ACCESS4 state. If $\overline{BLAST}$ is asserted, the machine proceeds to the IDLE state; otherwise, it proceeds to the ACCESS4 state.

The ACCESS4 state is the second data cycle (T_{d1}) for read accesses. In the middle of ACCESS4, $\overline{CASOO}$ is deasserted. At the end of ACCESS4, $\overline{CASOB}$3:0 are deasserted if they were earlier asserted. $\overline{CASOB}$3:0 are deasserted because $\overline{CASOO}$ is sampled deasserted. $\overline{CASOO}$ is reasserted at the end of ACCESS4 to initiate the fourth data (T_{d3}) access. From ACCESS4, the machine can proceed to either the IDLE state or the ACCESS5 state. If $\overline{BLAST}$ is asserted, the machine proceeds to the IDLE state; otherwise, to the ACCESS5 state.

The ACCESS5 state is the third data cycle (T_{d2}) for read accesses. In the middle of ACCESS5, $\overline{CASOB}$3:0 are asserted when $\overline{CASOO}$ and the respective byte enable signals from the processor are sampled asserted, whereas $\overline{CASEE}$ is deasserted. At the end of ACCESS5, the $\overline{CASEB}$3:0 are deasserted if they were earlier asserted. $\overline{CASEB}$3:0 are deasserted because $\overline{CASEE}$ is sampled deasserted. From ACCESS5, the machine can proceed to either the IDLE state or the ACCESS6 state. When $\overline{BLAST}$ is asserted, the machine proceeds to the IDLE state, otherwise to the ACCESS6 state.

ACCESS6 is the fourth and last data cycle (T_{d3}) for read accesses. In the middle of ACCESS6, $\overline{CASOO}$ is deasserted. At the end of ACCESS6, $\overline{CASOB}$3:0 are deasserted. $\overline{CASOB}$3:0 are deasserted because $\overline{CASOO}$ is sampled deasserted. From ACCESS6, the machine proceeds to IDLE state while deasserting $\overline{RASE}$ and $\overline{RASO}$.

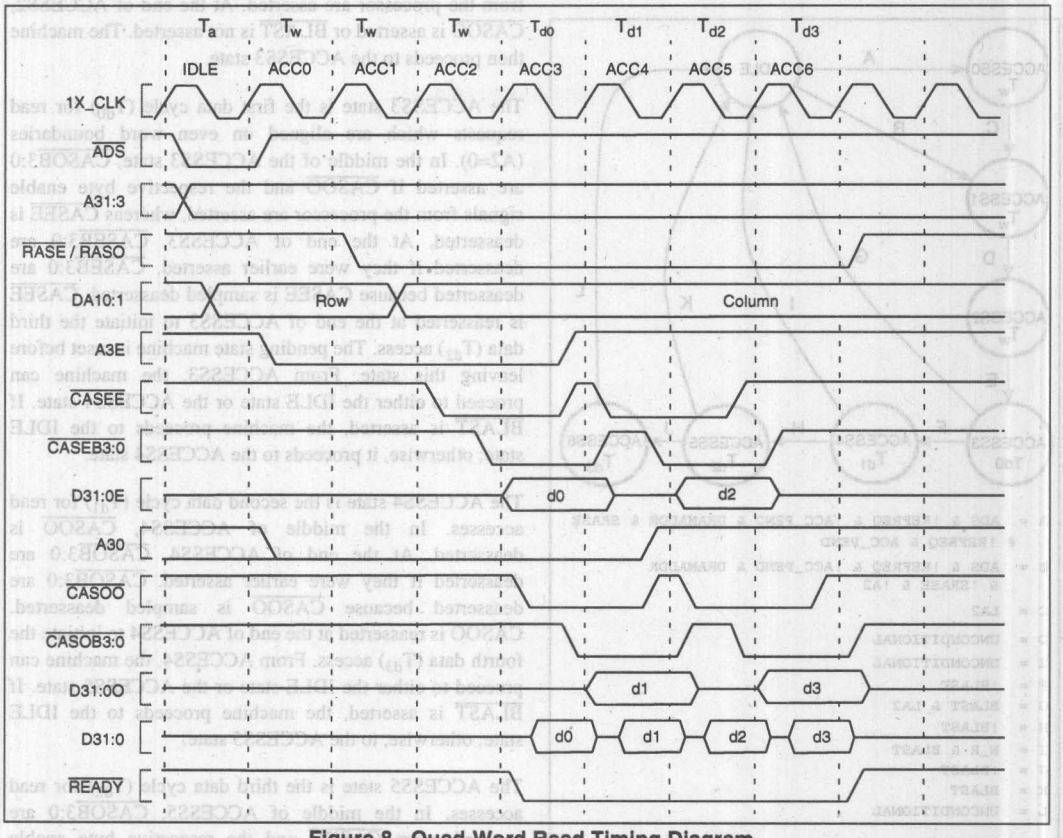


Figure 8. Quad-Word Read Timing Diagram

5.2 Single-Word Read

The ACCESS state machine takes a slightly different path when a read request is aligned on an odd word boundary. Figure 9 shows the state diagram for a single-word read; Figure 10 shows a single-word read timing diagram.

From the IDLE state, the machine enters the ACCESS0 state due to a processor request or a pending processor request. At the end of the IDLE state, the A3EVEN and A3ODD state machines are loaded with the processor's address A3 and the ODDACCESS state machine is loaded with the processor's address A2. $\overline{MUX}$ is deasserted in the IDLE state, which selects the row address.

At the end of the ACCESS0 state, $\overline{RASO}$ is asserted. The machine then proceeds to the ACCESS2 state.

$\overline{MUX}$ is asserted in the middle of the ACCESS2 state; this selects the column address. At the end of ACCESS2, $\overline{CASOO}$ is asserted. From ACCESS2, the machine enters ACCESS3 state.

In the middle of the ACCESS3 state, $\overline{CASOB}3:0$ are asserted when $\overline{CASOO}$ and the respective byte enable signals from the processor are asserted. The pending state machine is reset before leaving this state. The machine then proceeds to ACCESS4 state.

The ACCESS4 state is the data cycle (T_{d0}) for the read access. In the middle of ACCESS4, $\overline{CASOO}$ is deasserted. At the end of ACCESS4, $\overline{CASOB}3:0$ are deasserted. $\overline{CASOB}3:0$ are deasserted because $\overline{CASOO}$ is sampled deasserted. The machine then proceeds to the IDLE state.

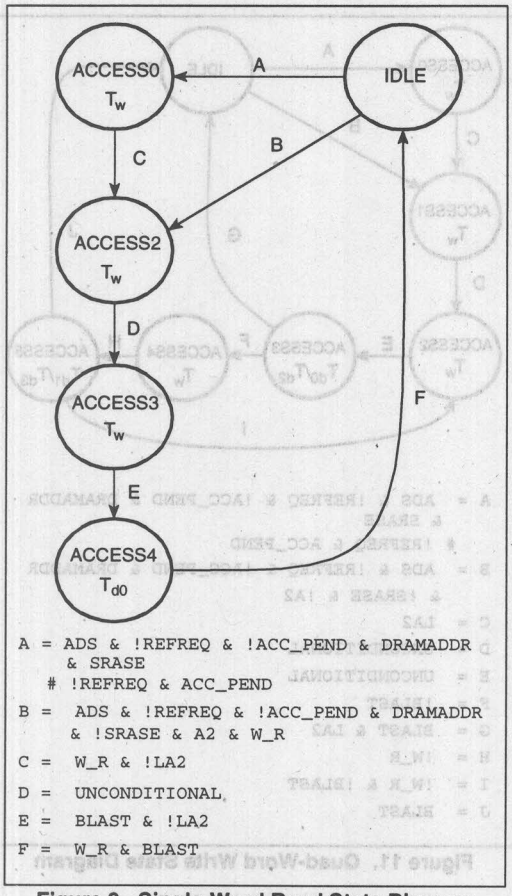


Figure 9. Single-Word Read State Diagram (A2 = 1)

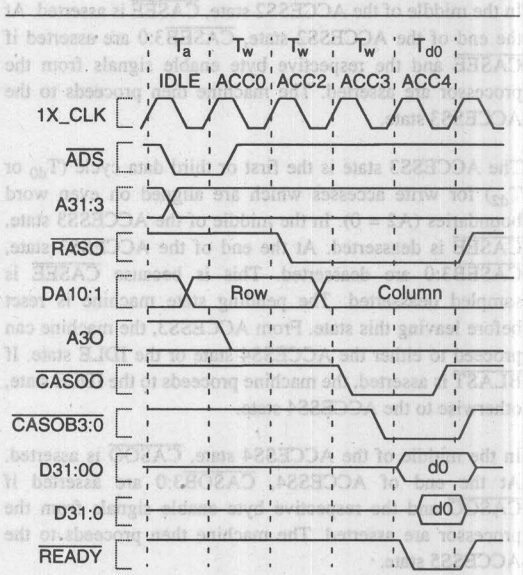


Figure 10. Single-Word Read Timing Diagram (A2 = 1)

5.3 Quad-Word Write

Figure 11 shows the state diagram for a quad-word write. This state diagram also shows the state machine paths for triple-, double-, and single-word writes. Single-word writes which are aligned on odd word boundaries use a different path; therefore, a different example is used to explain the state machine path. Figure 12 shows the timing diagram for a quad-word write.

From the IDLE state, the machine enters the ACCESS0 state due to a processor request or a pending processor request. At the end of the IDLE state, the A3EVEN and A3ODD state machines are loaded with the processor's address A3, and the ODDACCESS state machine is loaded with the processor's address A2. $\overline{MUX}$ is deasserted in the IDLE state, which selects the row address.

At the end of the ACCESS0 state, $\overline{RASE}$ and $\overline{RASO}$ are asserted. The machine then proceeds to the ACCESS1 state.

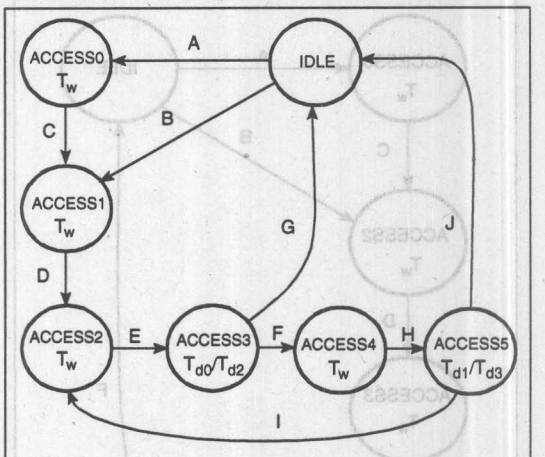
$\overline{MUX}$ is asserted in the middle of the ACCESS1 state; this selects the column address. From ACCESS1, the machine enters the ACCESS2 state.

In the middle of the ACCESS2 state, $\overline{CASEE}$ is asserted. At the end of the ACCESS2 state, $\overline{CASEB}$3:0 are asserted if $\overline{CASEE}$ and the respective byte enable signals from the processor are asserted. The machine then proceeds to the ACCESS3 state.

The ACCESS3 state is the first or third data cycle (T_{d0} or T_{d2}) for write accesses which are aligned on even word boundaries (A2 = 0). In the middle of the ACCESS3 state, $\overline{CASEE}$ is deasserted. At the end of the ACCESS3 state, $\overline{CASEB}$3:0 are deasserted. This is because $\overline{CASEE}$ is sampled deasserted. The pending state machine is reset before leaving this state. From ACCESS3, the machine can proceed to either the ACCESS4 state or the IDLE state. If $\overline{BLAST}$ is asserted, the machine proceeds to the IDLE state, otherwise to the ACCESS4 state.

In the middle of the ACCESS4 state, $\overline{CASOO}$ is asserted. At the end of ACCESS4, $\overline{CASOB}$3:0 are asserted if $\overline{CASOO}$ and the respective byte enable signals from the processor are asserted. The machine then proceeds to the ACCESS5 state.

The ACCESS5 state is the second or fourth data cycle (T_{d1} or T_{d3}) for write accesses which are aligned on even word boundary (A2 = 0). In the middle of ACCESS5, $\overline{CASOO}$ is deasserted. At the end of ACCESS5, $\overline{CASOB}$3:0 are deasserted. This is because $\overline{CASOO}$ is sampled deasserted. From ACCESS5, the machine can proceed to either the ACCESS2 state or the IDLE state. If $\overline{BLAST}$ is asserted, the machine proceeds to the IDLE state, otherwise to the ACCESS2 state.




```
A =  ADS & !REFREQ & !ACC_PEND & DRAMADDR
     & SRASE
   # !REFREQ & ACC_PEND
B =  ADS & !REFREQ & !ACC_PEND & DRAMADDR
     & !SRASE & !A2
C =  LA2
D =  UNCONDITIONAL
E =  UNCONDITIONAL
F =  !BLAST
G =  BLAST & LA2
H =  !W_R
I =  !W_R & !BLAST
J =  BLAST
```

Figure 11. Quad-Word Write State Diagram

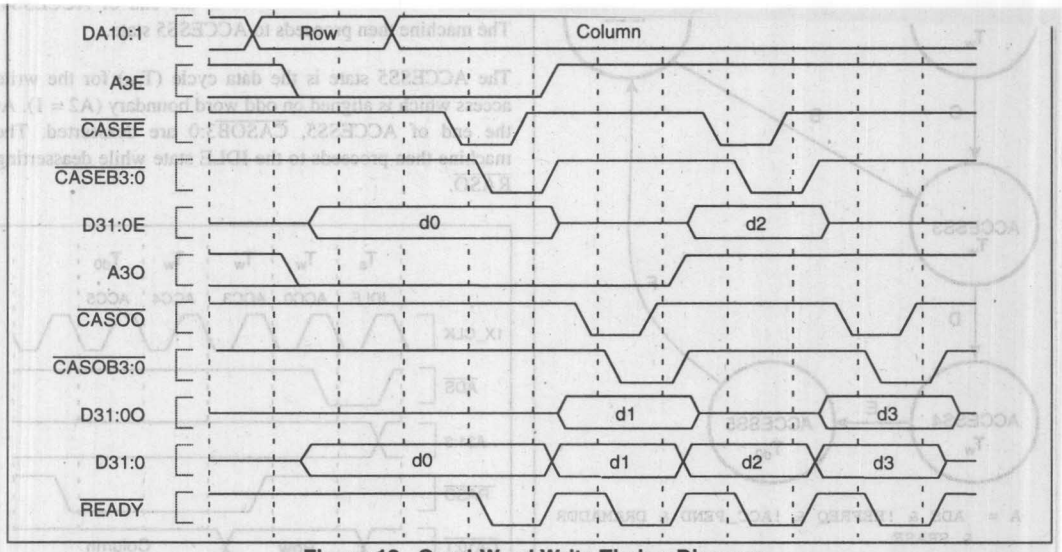


Figure 12. Quad-Word Write Timing Diagram

5.4 Single-Word Write

The ACCESS state machine takes a slightly different path when the write request is aligned on an odd word boundary. Figure 13 shows the state diagram for a single-word write. Figure 14 shows the timing diagram.

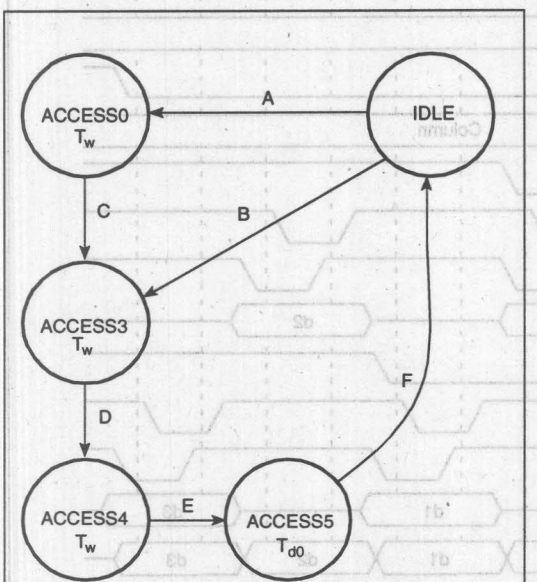


Figure 13. Single-Word Write State Diagram (A2 = 1)

From the IDLE state, the machine enters the ACCESS0 state due to a processor request or a pending processor request. At the end of the IDLE state, the A3EVEN and A3ODD state machines are loaded with the processor's address A3, and the ODDACCESS state machine is loaded with the processor's address A2.

At the end of ACCESS0 state, $\overline{RASO}$ is asserted. The machine then proceeds to the ACCESS3 state.

In the middle of ACCESS3, $\overline{MUX}$ is asserted. This causes the column address to be selected. From ACCESS3, the machine enters ACCESS4 state.

In the middle of ACCESS4, $\overline{CASOO}$ is asserted. At the end of the ACCESS4 state, $\overline{CASO}$3:0 are asserted if $\overline{CASOO}$ and the byte enable signals from the processor are sampled asserted. $\overline{CASOO}$ is deasserted at the end of ACCESS4. The machine then proceeds to ACCESS5 state.

The ACCESS5 state is the data cycle (T_{d0}) for the write access which is aligned on odd word boundary (A2 = 1). At the end of ACCESS5, $\overline{CASOB}$3:0 are deasserted. The machine then proceeds to the IDLE state while deasserting $\overline{RASO}$.

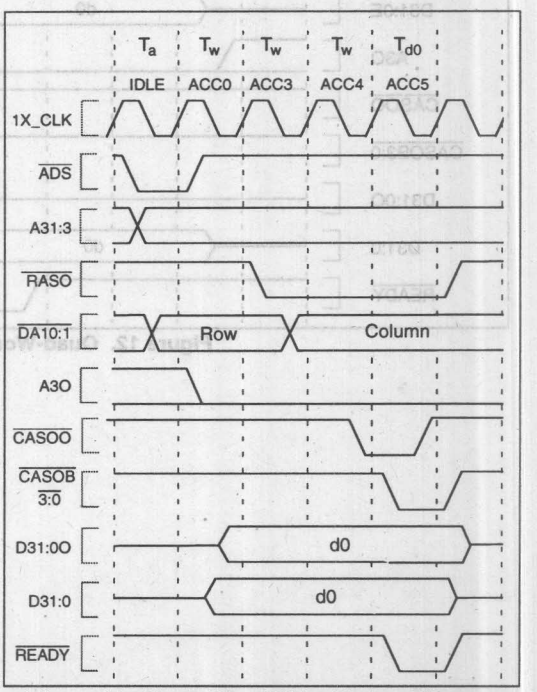


Figure 14. Single-Word Write Timing Diagram (A2 = 1)

5.5 Refresh Cycles

The refresh counter requests a DRAM refresh every 7.8 µs. One bank is refreshed at a time in alternation. The ACCESS state machine sequences the refresh and, based on the state of the RFEVENBK state machine, either the even or the odd bank is refreshed. The following text assumes the even bank is to be refreshed; for example, the RFEVENBK state machine is active. The odd bank is refreshed in a similar manner.

Figure 15 shows the refresh state diagram. Figure 16 shows the refresh timing diagram. From the IDLE state, the machine enters the REFRESH0 state if $\overline{\text{REFREQ}}$ is asserted. In the middle of REFRESH0, $\overline{\text{CASEE}}$ is asserted. At the end of REFRESH0, $\overline{\text{CASEB}}$3:0 is asserted because $\overline{\text{CASEE}}$ is sampled asserted. At the end of REFRESH0 the counter is also reloaded which deasserts $\overline{\text{REFREQ}}$ to get deasserted. Counting resumes on the next clock edge. The machine then proceeds to the REFRESH1 state.

In the middle of the REFRESH1 state, $\overline{\text{CASEE}}$ is deasserted while $\overline{\text{RASE}}$ is asserted. At the end of the REFRESH1 state, $\overline{\text{CASEB}}$3:0 are deasserted. This is because $\overline{\text{CASEE}}$ is sampled deasserted. The RFEVENBK state machine is toggled at the end of REFRESH1. The machine then proceeds to the REFRESH2 state. The next refresh sequence refreshes the odd bank because the state of the RFEVENBK state machine is altered.

In the REFRESH2 state, the machine unconditionally proceeds to the REFRESH3 state.

At the end of the REFRESH3 state, $\overline{\text{RASE}}$ is deasserted. The machine then proceeds to the IDLE state.

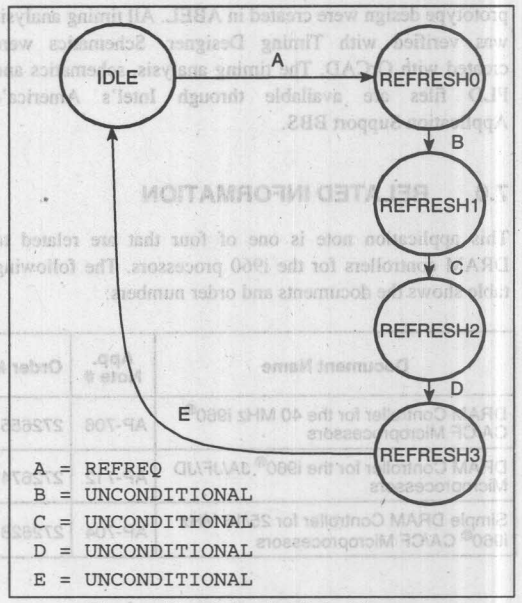


Figure 15. Refresh State Diagram

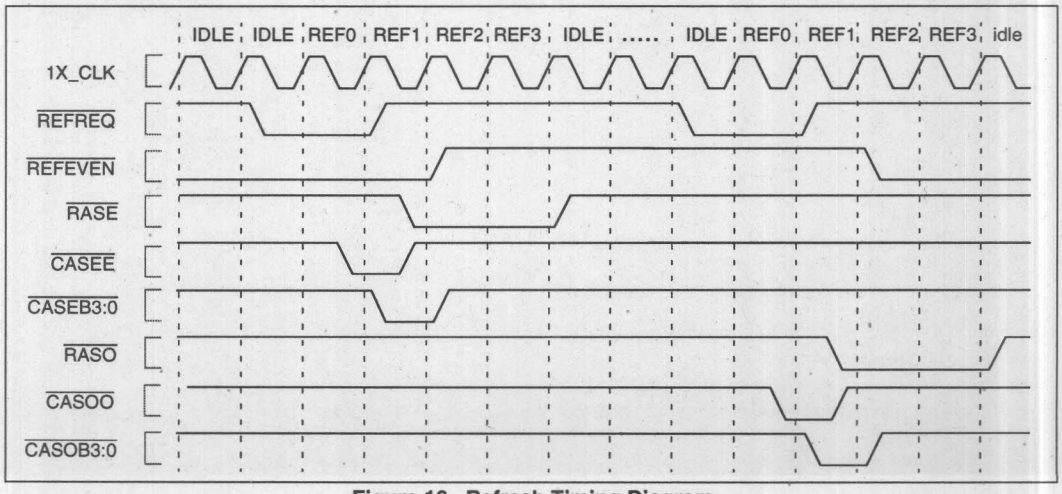


Figure 16. Refresh Timing Diagram

6.0 CONCLUSION

In conclusion, this application note describes a DRAM controller for use with 33 MHz i960 CA/CF processors. This DRAM controller was built and tested for validation purposes. The PLD equations used to build and test the prototype design were created in ABEL. All timing analysis was verified with Timing Designer. Schematics were created with OrCAD. The timing analysis, schematics and PLD files are available through Intel's America's Application Support BBS.

7.0 RELATED INFORMATION

This application note is one of four that are related to DRAM controllers for the i960 processors. The following table shows the documents and order numbers:

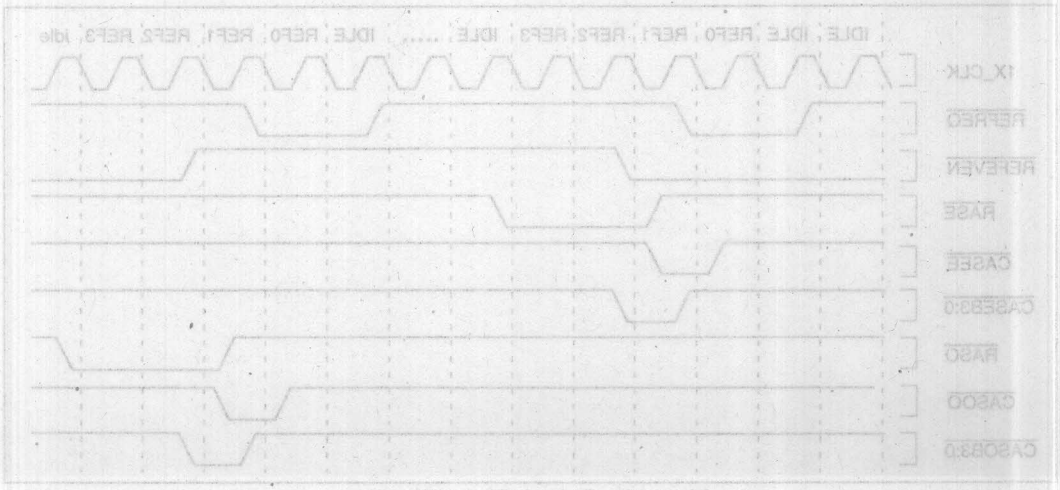
Document Name	App. Note #	Order #
DRAM Controller for the 40 MHz i960® CA/CF Microprocessors	AP-706	272655
DRAM Controller for the i960® JA/JF/JD Microprocessors	AP-712	272674
Simple DRAM Controller for 25/16 MHz i960® CA/CF Microprocessors	AP-704	272628

To receive these documents or any other available Intel literature, contact:

Intel Corporation
Literature Sales
P.O. Box 7641
Mt. Prospect IL 60056-7641
1-800-879-4683

To receive files that contain the timing analysis, schematics and PLD equations for this and the other DRAM controller application notes, contact:

Intel Corporation
America's Application Support BBS
916-356-3600



APPENDIX A
PLD EQUATIONS

Table A-1 contains the PLD equations which were used to build and test the prototype design. Table A-2 defines signal and product term allocation. The PLD equations were created in ABEL as a device-independent design. Using the ABEL* software, a PDS file was created and subsequently imported into PLDSHELL* software. PLDSHELL was used to fit the design into an Altera EPX780 FLEXlogic* PLD. PLDSHELL was also utilized to create the JEDEC file, and to simulate the design.

In addition, this appendix contains a table listing the number of product terms used by each macrocell.

The DRAM Controller does not use the $\overline{\text{APK\_ACTIVE}}$ signal.

Table A-1. 33 MHz DRAM Controller PLD Equations (Sheet 1 of 22)

```
Module          CX33T
Title           'DRAM Controller for 25/33MHz
Source File     CX33T.ABL
Revision        Rev 0.0
Date            11/17/94
Designer        Sailesh Bissessur
                Intel 80960 Applications Engineering'
"-------------------------------------------------------------------
" 2-Way Interleaved DRAM controller for the 960Cx 33 AND 25MHz.
" This design also contains logic for FLASH, HEX DISPLAY, and Software Reset.
"       DRAM            -  0xA0000000
"       FLASH           -  0xFFFE0000
"       HEX DISPLAY     -  0xB8000000
"       SW Reset        -  0xB0000000
"-------------------------------------------------------------------
"       Uxx    device  'IFX780_132';
" inputs signals
"
        CLK1            PIN 118 ; " 1x clock
        CLK2            PIN 52  ; " 2x clock

        A2              PIN; " Address A2
        !EXTRST         PIN; " External Reset
        !CPUWAIT        PIN; " Processor wait
        !ADS            PIN; " Address Strobe
        !BLAST          PIN; " Burst Last
        !W_R            PIN; " Processor Read/Write
        A31             PIN; " Address A31
        A30             PIN; " Address A30
        A29             PIN; " Address A29
        A28             PIN; " Address A28
        A27             PIN; " Address A27
        DCLK1           PIN; " Delayed Clock
        A3              PIN; " Address A3
        !BE3            PIN; " Byte Enable 3
        !BE2            PIN; " Byte Enable 2
        !BE1            PIN; " Byte Enable 1
        !BE0            PIN; " Byte Enable 0
        !APK_ACTIVE     PIN; " Indicates presence of ApLink
```

Table A-1. 33 MHz DRAM Controller PLD Equations (Sheet 2 of 22)

```
" output signals
        !LA2            PIN  istype 'reg'; " latched A2
        Q3              PIN  istype 'reg'; " m/c bit3
        Q2              PIN  istype 'reg'; " m/c bit2
        !RDEN           PIN  istype 'com'; " enables data mux '257s
        !SELA           PIN  istype 'reg'; " selects even odd data
        !SELB           PIN  istype 'reg'; " selects even odd data
        !READY          PIN  istype 'reg'; " Processor READY
        Q1              PIN  istype 'reg'; " m/c bit1
        Q0              PIN  istype 'reg'; " m/c bit0
        !ACC_PEND       PIN  istype 'reg'; " access pending indicator
        !RASO           PIN  istype 'reg'; " Odd RAS
        A3E             PIN  istype 'reg'; " Even Address Counter
        !REFEVEN        PIN  istype 'reg'; " which bank to refresh
        !CASEE          PIN  istype 'reg'; " Pipelined Even CAS
        !CASOO          PIN  istype 'reg'; " Pipelined Odd CAS
        A3O             PIN  istype 'reg'; " Odd Address Counter
        WAIT            PIN  istype 'com'; " wait state indicator
        !RASE           PIN  istype 'reg'; " Even RAS
        !MUX            PIN  istype 'reg'; " Selects between Even/Odd Col Addr

        !CASEB0         PIN  istype 'reg'; " Byte 0 Even CAS
        !CASEB1         PIN  istype 'reg'; " Byte 1 Even CAS
        !CASEB2         PIN  istype 'reg'; " Byte 2 Even CAS
        !CASEB3         PIN  istype 'reg'; " Byte 3 Even CAS
        !CASOB0         PIN  istype 'reg'; " Byte 0 Odd CAS
        !CASOB1         PIN  istype 'reg'; " Byte 1 Odd CAS
        !CASOB2         PIN  istype 'reg'; " Byte 2 Odd CAS
        !CASOB3         PIN  istype 'reg'; " Byte 3 Odd CAS
        S3              PIN  istype 'reg'; " Refresh Counter 1 bit 3
        S2              PIN  istype 'reg'; " Refresh Counter 1 bit 2
        S1              PIN  istype 'reg'; " Refresh Counter 1 bit 1
        S0              PIN  istype 'reg'; " Refresh Counter 1 bit 0
        T3              PIN  istype 'reg'; " Refresh Counter 2 bit 3
        T2              PIN  istype 'reg'; " Refresh Counter 2 bit 2
        T1              PIN  istype 'reg'; " Refresh Coubter 2 bit 1
        T0              PIN  istype 'reg'; " Refresh Counter 2 bit 0
        !REFREQ         PIN  istype 'com'; " Refresh Required
        !FLASHCS        PIN  istype 'reg'; " FLASH Chip Select
        !FLASHRD        PIN  istype 'reg'; " FLASH OE
        !FLASHWR        PIN  istype 'com'; " Flash WE
        !XCROE          PIN  istype 'reg'; " TRANSCEIVER OE control
        !XCRDIR         PIN  istype 'com'; " TRANSCEIVER DIR control
        !SWRST          PIN  istype 'reg'; " SW Reset Indicator
        !TRIGRST        PIN  istype 'reg'; " Triggers the Reset Device
        !RESET          PIN  istype 'reg'; " System Reset
        !WRE            PIN  istype 'com'; " Even Bank WE
        !WRO            PIN  istype 'com'; " Odd Bank WE
        !SRASE          PIN  istype 'reg'; " Latched RASE or RASO
        LED_LAT         PIN  istype 'reg'; " HEX Display Pulse
```

Table A-1. 33 MHz DRAM Controller PLD Equations (Sheet 3 of 22)

```
"
C = .C.;
X = .X.;
"
CYCLE        = [Q3,Q2,Q1,Q0];
ODDACCESS    = [LA2];
BANKSELA     = [SELA];
BANKSELB     = [SELB];
PENDING      = [ACC_PEND];
RDY          = [READY];
RASEVEN      = [RASE];
RASODD       = [RASO];
CASPIPE      = [CASEE];
CASPIPO      = [CASOO];
ADDRMUX      = [MUX];
A3EVEN       = [A3E];
A3ODD        = [A3O];
RFEVENBK     = [REFEVEN];
CASE_B0      = [CASEB0];
CASE_B1      = [CASEB1];
CASE_B2      = [CASEB2];
CASE_B3      = [CASEB3];
CASO_B0      = [CASOB0];
CASO_B1      = [CASOB1];
CASO_B2      = [CASOB2];
CASO_B3      = [CASOB3];
REFCT2 = [T3,T2,T1,T0];
REFCT1 = [S3,S2,S1,S0];
DRAMADDR  = (A31 & !A30 &  A29 & !A28 & !A27);
FLASHADDR = (A31 &  A30 &  A29 &  A28 &  A27);
SWRSTADDR = (A31 & !A30 &  A29 &  A28 & !A27);
LEDADDR   = (A31 & !A30 &  A29 &  A28 &  A27);

"
ASSERT   = ^b1;
DEASSERT = ^b0;

Z0       = ^b0000;
Z1       = ^b0001;
Z2       = ^b0010;
Z3       = ^b0011;
Z4       = ^b0100;
Z5       = ^b0101;
Z6       = ^b0110;
Z7       = ^b0111;
Z8       = ^b1000;
Z9       = ^b1001;
Z10      = ^b1010;
Z11      = ^b1011;
Z12      = ^b1100;
Z13      = ^b1101;
Z14      = ^b1110;
Z15      = ^b1111;
```

Table A-1. 33 MHz DRAM Controller PLD Equations (Sheet 4 of 22)

```
IDLE          = ^b0000;
ACCESS0       = ^b0001;
ACCESS1       = ^b0010;
ACCESS2       = ^b0011;
ACCESS3       = ^b0100;
ACCESS4       = ^b0101;
ACCESS5       = ^b0110;
ACCESS6       = ^b0111;
ACCESS7       = ^b1000; "this state is never entered
ACCESS8       = ^b1001; "this state is never entered
REFRESH0      = ^b1010;
REFRESH1      = ^b1011;
REFRESH2      = ^b1100;
REFRESH3      = ^b1101;
REFRESH4      = ^b1110; "this state is never entered
REFRESH5      = ^b1111; "this state is never entered

"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"Holds state of A2 of Processor
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
state_diagram ODDACCESS
                state ASSERT:
                        if((CYCLE == IDLE) & A2) then DEASSERT
                else
                        ASSERT;

                state DEASSERT:
                        if((CYCLE == IDLE) & !A2) then ASSERT
                else
                        DEASSERT;
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"Even byte 0 CAS
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
state_diagram CASE_B0
                state ASSERT:
                        if(!Q3 & !CASEE) then DEASSERT
                else
                        if(!Q3 & !WAIT & BLAST & W_R & !DCLK1) then DEASSERT
                else
                        if(Q3 & !CASEE) then DEASSERT
                else
                        ASSERT;

                state DEASSERT:
                        if(!Q3 & CASEE & BE0) then ASSERT
                else
                        if(Q3 & CASEE) then ASSERT
                else
                        DEASSERT;
```

Table A-1. 33 MHz DRAM Controller PLD Equations (Sheet 5 of 22)

```
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"Even byte 1 CAS
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
state_diagram CASE_B1
                state ASSERT:
                        if(!Q3 & !CASEE) then DEASSERT
                else
                        if(!Q3 & !WAIT & BLAST & W_R & !DCLK1) then DEASSERT
                else
                        if(Q3 & !CASEE) then DEASSERT
                else
                        ASSERT;

                state DEASSERT:
                        if(!Q3 & CASEE & BE1) then ASSERT
                else
                        if(Q3 & CASEE) then ASSERT
                else
                        DEASSERT;
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"Even byte 2 CAS
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
state_diagram CASE_B2
                state ASSERT:
                        if(!Q3 & !CASEE) then DEASSERT
                else
                        if(!Q3 & !WAIT & BLAST & W_R & !DCLK1) then DEASSERT
                else
                        if(Q3 & !CASEE) then DEASSERT
                else
                        ASSERT;

                state DEASSERT:
                        if(!Q3 & CASEE & BE2) then ASSERT
                else
                        if(Q3 & CASEE) then ASSERT
                else
                        DEASSERT;
```

Table A-1. 33 MHz DRAM Controller PLD Equations (Sheet 6 of 22)

```
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"Even byte 3 CAS
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
state_diagram CASE_B3
                state ASSERT:
                        if(!Q3 & !CASEE) then DEASSERT
                else
                        if(!Q3 & !WAIT & BLAST & W_R & !DCLK1) then DEASSERT
                else
                        if(Q3 & !CASEE) then DEASSERT
                else
                        ASSERT;

                state DEASSERT:
                        if(!Q3 & CASEE & BE3) then ASSERT
                else
                        if(Q3 & CASEE) then ASSERT
                else
                        DEASSERT;
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"Odd byte 0 CAS
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
state_diagram CAS0_B0
                state ASSERT:
                        if(!Q3 & !CAS00) then DEASSERT
                else
                        if(!Q3 & !WAIT & BLAST & W_R & !DCLK1) then DEASSERT
                else
                        if(Q3 & !CAS00) then DEASSERT
                else
                        ASSERT;

                state DEASSERT:
                        if(!Q3 & CAS00 & BE0) then ASSERT
                else
                        if(Q3 & CAS00) then ASSERT
                else
                        DEASSERT;
```

Table A-1. 33 MHz DRAM Controller PLD Equations (Sheet 7 of 22)

```
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"Odd byte 1 CAS
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
state_diagram CASO_B1
                state ASSERT:
                        if(!Q3 & !CASOO) then DEASSERT
                else
                        if(!Q3 & !WAIT & BLAST & W_R & !DCLK1) then DEASSERT
                else
                        if(Q3 & !CASOO) then DEASSERT
                else
                        ASSERT;

                state DEASSERT:
                        if(!Q3 & CASOO & BE1) then ASSERT
                else
                        if(Q3 & CASOO) then ASSERT
                else
                        DEASSERT;
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"Odd byte 2 CAS
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
state_diagram CASO_B2
                state ASSERT:
                        if(!Q3 & !CASOO) then DEASSERT
                else
                        if(!Q3 & !WAIT & BLAST & W_R & !DCLK1) then DEASSERT
                else
                        if(Q3 & !CASOO) then DEASSERT
                else
                        ASSERT;

                state DEASSERT:
                        if(!Q3 & CASOO & BE2) then ASSERT
                else
                        if(Q3 & CASOO) then ASSERT
                else
                        DEASSERT;
```

```
                else
                        ASSERT;

                state DEASSERT:
                        if(!Q3 & CASOO & BE3) then ASSERT
                else
                        if(Q3 & CASOO) then ASSERT
                else
                        DEASSERT;
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"Keeps track of any pending access
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
state_diagram PENDING
                state ASSERT:
                        if(CYCLE == ACCESS3) then DEASSERT
                else
                        ASSERT;

                state DEASSERT:
                        if(ADS & DRAMADDR) then ASSERT
                else
                        DEASSERT;
```

Table A-1. 33 MHz DRAM Controller PLD Equations (Sheet 9 of 22)

```
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"Indicates which Bank is to be refreshed next when !REFREQ becomes active
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
state_diagram RFEVENBK
                state ASSERT:
                        if((CYCLE == REFRESH1) & !DCLK1) then DEASSERT
                else
                        ASSERT;

                state DEASSERT:
                        if((CYCLE == REFRESH1) & !DCLK1) then ASSERT
                else
                        DEASSERT;
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"Selects even or odd data path while reading
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
state_diagram BANKSELA
                state ASSERT:
                        if((CYCLE == IDLE) & A2) then DEASSERT
                else
                        if((CYCLE == ACCESS3)) then DEASSERT
                else
                        if((CYCLE == ACCESS5)) then DEASSERT
                else
                        ASSERT;

                state DEASSERT:
                        if((CYCLE == IDLE) & !A2) then ASSERT
                else
                        if((CYCLE == ACCESS4)) then ASSERT
                else
                        DEASSERT;
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"Selects even or odd data path while reading
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
state_diagram BANKSELB
                state ASSERT:
                        if((CYCLE == IDLE) & A2) then DEASSERT
                else
                        if((CYCLE == ACCESS3)) then DEASSERT
                else
                        if((CYCLE == ACCESS5)) then DEASSERT
                else
                        ASSERT;

                state DEASSERT:
                        if((CYCLE == IDLE) & !A2) then ASSERT
                else
                        if((CYCLE == ACCESS4)) then ASSERT
                else
                        DEASSERT;
```

Table A-1. 33 MHz DRAM Controller PLD Equations (Sheet 10 of 22)

```
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"Generates READY to the processor
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
state_diagram RDY
                state ASSERT:
                        if(W_R & BLAST) then DEASSERT
                else
                        if(!W_R) then DEASSERT
                else
                        ASSERT;

                state DEASSERT:
                        if((CYCLE == ACCESS2) & LA2) then ASSERT
                else
                        if((CYCLE == ACCESS3) & W_R & !LA2) then ASSERT
                else
                        if((CYCLE == ACCESS3) & W_R & LA2 & !BLAST) then ASSERT
                else
                        if((CYCLE == ACCESS4) & !W_R) then ASSERT
                else
                        DEASSERT;
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"Even RAS
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
state_diagram RASEVEN
                state ASSERT:
                     if((CYCLE == ACCESS3) & !DCLK1 & LA2 & BLAST) then DEASSERT
                else
                     if((CYCLE == ACCESS4) & !DCLK1 & W_R & BLAST) then DEASSERT
                else
                      if((CYCLE == ACCESS5) & !DCLK1 & BLAST) then DEASSERT
                else
                      if((CYCLE == ACCESS6) & !DCLK1 & BLAST) then DEASSERT
                else
                      if((CYCLE == REFRESH3) & !DCLK1) then DEASSERT
                else
                       ASSERT;

                state DEASSERT:
                      if((CYCLE == IDLE) & !SRASE & ADS & !REFREQ & !ACC_PEND
                         & DRAMADDR & !DCLK1) then ASSERT
                else
                      if((CYCLE == ACCESS0) & !DCLK1) then ASSERT
                else
                      if((CYCLE ==  REFRESH1) & DCLK1 & REFEVEN) then ASSERT
                else
                       DEASSERT;
```

Table A-1. 33 MHz DRAM Controller PLD Equations (Sheet 11 of 22)

```
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"Odd RAS
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
state_diagram RASODD
                state ASSERT:
                      if((CYCLE == ACCESS3) & !DCLK1 & LA2 & BLAST) then DEASSERT
                else
                      if((CYCLE == ACCESS4) & !DCLK1 & W_R & BLAST) then DEASSERT
                else
                       if((CYCLE == ACCESS5) & !DCLK1 & BLAST) then DEASSERT
                else
                       if((CYCLE == ACCESS6) & !DCLK1 & BLAST) then DEASSERT
                else
                       if((CYCLE == REFRESH3) & !DCLK1) then DEASSERT
                else
                        ASSERT;

                state DEASSERT:
                       if((CYCLE == IDLE) & !SRASE & ADS & !REFREQ & !ACC_PEND
                          & DRAMADDR & !DCLK1) then ASSERT
                else
                       if((CYCLE == ACCESS0)  & !DCLK1)  then ASSERT
                else
                       if((CYCLE == REFRESH1) &  DCLK1 & !REFEVEN) then ASSERT
                else
                        DEASSERT;
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"Pipelined Even CAS
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
state_diagram CASPIPE
                state ASSERT:
                       if((CYCLE == ACCESS3) & DCLK1) then DEASSERT
                else
                       if((CYCLE == ACCESS5) & DCLK1) then DEASSERT
                else
                      if((CYCLE == ACCESS4) & !DCLK1 & BLAST & W_R) then DEASSERT
                else
                       if((CYCLE == REFRESH1) & DCLK1) then DEASSERT
                else
                        ASSERT;

                state DEASSERT:
                       if((CYCLE == ACCESS1) & W_R & !DCLK1) then ASSERT
                else
                       if((CYCLE == ACCESS2) & !W_R & DCLK1) then ASSERT
                else
                       if((CYCLE == ACCESS3) & W_R & !DCLK1 & !BLAST) then ASSERT
                else
                       if((CYCLE == REFRESH0) & DCLK1 & REFEVEN) then ASSERT
                else
                        DEASSERT;
```

```
                else
                        if((CYCLE == ACCESS5) & !W_R & DCLK1) then DEASSERT
                else
                        if((CYCLE == REFRESH1) & DCLK1) then DEASSERT
                else
                         ASSERT;

                state DEASSERT:
                     if((CYCLE == ACCESS2) & W_R & LA2 & !BLAST & !DCLK1) then
                          ASSERT
                else
                        if((CYCLE == ACCESS2) & W_R & !LA2 & !DCLK1) then ASSERT
                else
                        if((CYCLE == ACCESS4) & W_R & !BLAST & !DCLK1) then ASSERT
                else
                        if((CYCLE == ACCESS4) & !W_R & DCLK1) then ASSERT
                else
                        if((CYCLE == REFRESH0) & DCLK1 & !REFEVEN) then ASSERT
                else
                         DEASSERT;
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"Even address counter
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
state_diagram A3EVEN
                state DEASSERT:
                        if((CYCLE == IDLE) & !DCLK1 &  A3) then ASSERT
                else
                        if((CYCLE == ACCESS3) & !W_R & !DCLK1) then ASSERT
                else
                        if((CYCLE == ACCESS3) &  W_R & DCLK1) then ASSERT
                else
                        DEASSERT;

                state ASSERT:
                        if((CYCLE == IDLE) & !DCLK1 & !A3) then DEASSERT
                else
                        ASSERT;
```

Table A-1. 33 MHz DRAM Controller PLD Equations (Sheet 13 of 22)

```
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"Odd address counter
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
state_diagram A3ODD
                state DEASSERT:
                        if((CYCLE == IDLE) &  A3 & !DCLK1) then ASSERT
                else
                        if((CYCLE == ACCESS4) &  W_R & DCLK1) then ASSERT
                else
                        if((CYCLE == ACCESS5) & !W_R & !DCLK1) then ASSERT
                else
                        DEASSERT;

                state ASSERT:
                        if((CYCLE == IDLE) & !A3 & !DCLK1)  then DEASSERT
                else
                        ASSERT;
```

Table A-1. 33 MHz DRAM Controller PLD Equations (Sheet 14 of 22)

```
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"Main DRAM state machine - ACCESS state machine
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
state_diagram CYCLE
                state IDLE:
                        if(ADS & !REFREQ & !ACC_PEND & DRAMADDR & SRASE) then
                           ACCESS0
                else
                        if(ADS & !REFREQ & !ACC_PEND & DRAMADDR & !SRASE & !A2)
                           then ACCESS1
                else
                        if(ADS & !REFREQ & !ACC_PEND & DRAMADDR & !SRASE &  A2
                           & W_R) then ACCESS2
                else
                        if(ADS & !REFREQ & !ACC_PEND & DRAMADDR & !SRASE &  A2
                           & !W_R) then ACCESS3
                else
                        if(!REFREQ & ACC_PEND) then ACCESS0
                else
                        if(REFREQ) then REFRESH0
                else
                        IDLE;

                state ACCESS0:
                        if(W_R & !LA2) then ACCESS2
                else
                        if(!W_R & !LA2) then ACCESS3
                else
                        ACCESS1;

                state ACCESS1:
                        goto ACCESS2;

                state ACCESS2:
                        goto ACCESS3;

                state ACCESS3:
                        if(BLAST & LA2) then IDLE
                else
                        ACCESS4;

                state ACCESS4:
                        if(W_R & BLAST) then IDLE
                else
                        goto ACCESS5;

                state ACCESS5:
                        if(BLAST) then IDLE
                else
                        if(!W_R & !BLAST) then ACCESS2
                else
                        goto ACCESS6;
```

```
        state REFRESH1:
                goto REFRESH2;

        state REFRESH2:
                goto REFRESH3;

        state REFRESH3:
                goto IDLE;

        state REFRESH4:
                goto IDLE;

        state REFRESH5:
                goto IDLE;
```

Table A-1. 33 MHz DRAM Controller PLD Equations (Sheet 16 of 22)

```
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"Row/Column address select
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
state_diagram ADDRMUX
                state ASSERT:
                        if(!RASE & DCLK1) then DEASSERT
                else
                        ASSERT;

                state DEASSERT:
                        if(RASE & DCLK1) then ASSERT
                else
                        DEASSERT;
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"Refresh Counter 1
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
state_diagram REFCT1
                state Z0:
                        if(!Q3 & (REFCT2 == Z0)) then Z0
                else
                        Z15;

                state Z1:
                        if(Q3) then Z15
                else
                        Z0;

                state Z2:
                        if(Q3) then Z15
                else
                        Z1;

                state Z3:
                        if(Q3) then Z15
                else
                        Z2;

                state Z4:
                        if(Q3) then Z15
                else
                        Z3;

                state Z5:
                        if(Q3) then Z15
                else
                        Z4;

                state Z6:
                        if(Q3) then Z15
                else
                        Z5;
```

Table A-1. 33 MHz DRAM Controller PLD Equations (Sheet 17 of 22)

```
state Z7:
        if(Q3) then Z15
else
        Z6;

state Z8:
        if(Q3) then Z15
else
        Z7;

state Z9:
        if(Q3) then Z15
else
        Z8;

state Z10:
        if(Q3) then Z15
else
        Z9;

state Z11:
        if(Q3) then Z15
else
        Z10;

state Z12:
        if(Q3) then Z15
else
        Z11;

state Z13:
        if(Q3) then Z15
else
        Z12;

state Z14:
        if(Q3) then Z15
else
        Z13;

state Z15:
        if(Q3) then Z15
else
        Z14;
```

Table A-1. 33 MHz DRAM Controller PLD Equations (Sheet 18 of 22)

```
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"Refresh Counter 2
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
state_diagram REFCT2
                state Z0:
                        if(Q3) then Z15
                else
                        Z0;

                state Z1:
                        if(Q3) then Z15
                else
                        if(!Q3 & (REFCT1 == Z0)) then Z0;
                else
                        Z1;

                state Z2:
                        if(Q3) then Z15
                else
                        if(!Q3 & (REFCT1 == Z0)) then Z1;
                else
                        Z2;

                state Z3:
                        if(Q3) then Z15
                else
                        if(!Q3 & (REFCT1 == Z0)) then Z2;
                else
                        Z3;

                state Z4:
                        if(Q3) then Z15
                else
                        if(!Q3 & (REFCT1 == Z0)) then Z3;
                else
                        Z4;

                state Z5:
                        if(Q3) then Z15
                else
                        if(!Q3 & (REFCT1 == Z0)) then Z4;
                else
                        Z5;

                state Z6:
                        if(Q3) then Z15
                else
                        if(!Q3 & (REFCT1 == Z0)) then Z5;
                else
                        Z6;
```

```
state Z9:
        if(Q3) then Z15
else
        if(!Q3 & (REFCT1 == Z0)) then Z8;
else
        Z9;

state Z10:
        if(Q3) then Z15
else
        if(!Q3 & (REFCT1 == Z0)) then Z9;
else
        Z10;

state Z11:
        if(Q3) then Z15
else
        if(!Q3 & (REFCT1 == Z0)) then Z10;
else
        Z11;

state Z12:
        if(Q3) then Z15
else
        if(!Q3 & (REFCT1 == Z0)) then Z11;
else
        Z12;

state Z13:
        if(Q3) then Z15
else
        if(!Q3 & (REFCT1 == Z0)) then Z12;
else
        Z13;
```

Table A-1. 33 MHz DRAM Controller PLD Equations (Sheet 20 of 22)

```
state Z14:
        if(Q3) then Z15
else
        if(!Q3 & (REFCT1 == Z0)) then Z13;
else
        Z14;

state Z15:
        if(Q3) then Z15
else
        if(!Q3 & (REFCT1 == Z0)) then Z14;
else
        Z15;
```

Table A-1. 33 MHz DRAM Controller PLD Equations (Sheet 21 of 22)

```
"
"Equations
"
EQUATIONS
[Q3,Q2,Q1,Q0,!SELA,!SELB,!READY,!LA2,!ACC_PEND].clk = CLK1;
[Q3..Q0].RE = RESET;
[!LA2,!ACC_PEND,!READY,!SELA,!SELB].PR = RESET;
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"Indicates wait state cycles
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
WAIT =    (CYCLE == IDLE) & W_R
        # (CYCLE == ACCESS0) & W_R
        # (CYCLE == ACCESS1) & W_R
        # (CYCLE == ACCESS2) & W_R
        # (CYCLE == ACCESS3) & W_R & !LA2;

[!MUX,!RASE,!RASO,!CASEE,!CASOO,A3E,A3O,!REFEVEN].clk = CLK2;
[!MUX,!RASE,!RASO,!CASEE,!CASOO,A3E,A3O,!REFEVEN].pr = RESET;

[!CASEB0,!CASEB1,!CASEB2,!CASEB3,!CASOB0,!CASOB1,!CASOB2,!CASOB3].clk = CLK2;

[T3,T2,T1,T0,S3,S2,S1,S0].clk = CLK1;
[T3,T2,T1,T0,S3,S2,S1,S0].pr = RESET;

"
"Refresh required indicator
"
REFREQ = !T3 & !T2 & !T1 & !T0 & !S3 & !S2 & !S1 & !S0;
"
"FLASH Chip Select
"
FLASHCS :=   ADS & FLASHADDR & !APK_ACTIVE
           # !ADS & !BLAST & FLASHCS;
"
"FLASH OE control
"
FLASHRD = FLASHCS &  W_R;
"
"XCR OE control
"
XCROE :=  FLASHCS & !BLAST & !APK_ACTIVE
        # !ADS & LEDADDR & !BLAST;
"
"XCR DIR control
"
XCRDIR = W_R;
"
" Software reset indicator
"
SWRST :=    ADS & SWRSTADDR
          # !ADS & !BLAST & SWRST;
```

Table A-1. 33 MHz DRAM Controller PLD Equations (Sheet 22 of 22)

```
"
" Triggers the 7705 Reset Device
"
TRIGRST := TRIGRST;
TRIGRST.RE = SWRST;
"
"Pulse to the HEX DISPLAY
"
LED_LAT := !ADS & LEDADDR & XCROE & !BLAST;
"
"Latched RASE or RASO
"
SRASE := RASE # RASO;
"
"DRAM data path OE control while reading
"
RDEN = !Q3 &  W_R & RASE;
"
"Even DRAM data path control while writing
"
WRE =  !Q3 & !W_R & RASE;
"
"Odd DRAM data path control while writing
"
WRO =  !Q3 & !W_R & RASE;
[!FLASHCS,!XCROE,!SWRST,!TRIGRST,LED_LAT,SRASE].clk = CLK1;
[!FLASHCS,!XCROE,!SWRST,!TRIGRST].pr = RESET;
LED_LAT.RE = RESET;
"
"Latched external reset
"
RESET := EXTRST;
RESET.CLK = CLK1;
" Test vectors
end CX33T
```

A3E	4	$\overline{LA2}$	5
A3O	4	$\overline{WAIT}$	2
$\overline{RDEN}$	1	$\overline{REFEVEN}$	6
$\overline{SELA}$	4	T3	2
$\overline{MUX}$	2	T2	8
Q3	3	T1	7
Q2	8	T0	6
Q1	9	S3	5
Q0	8	S2	4
$\overline{CASEB3}$	3	S1	3
$\overline{CASEB2}$	3	S0	2
$\overline{CASEB1}$	3	$\overline{REFREQ}$	1
$\overline{CASEB0}$	3	$\overline{SWRST}$	2
$\overline{CASOB3}$	3	$\overline{SRASE}$	1
$\overline{CASOB2}$	3		
$\overline{CASOB1}$	3		
$\overline{CASOB0}$	3		
$\overline{FLASHCS}$	2		
$\overline{FLASHRD}$	1		
$\overline{TRIGRST}$	1		
$\overline{XCROE}$	2		
$\overline{XCRDIR}$	1		
$\overline{RESET}$	1		
$\overline{WRE}$	1		
$\overline{WRO}$	1		
LED_LAT	1		
$\overline{SELB}$	4		

Table A-2. Signal and Product Term Allocation

OUTPUT MACROCELLS		BURIED MACROCELLS	
Signal	Product Terms	Signal	Product Terms
$\overline{RASE}$	7	$\overline{CASEE}$	12
$\overline{RASO}$	7	$\overline{CASOO}$	14
$\overline{READY}$	5	$\overline{ACC\_PEND}$	5
A3E	4	$\overline{LA2}$	5
A3O	4	$\overline{WAIT}$	2
$\overline{RDEN}$	1	$\overline{REFEVEN}$	6
$\overline{SELA}$	4	T3	2
$\overline{MUX}$	2	T2	8
Q3	3	T1	7
Q2	8	T0	8
Q1	9	S3	5
Q0	8	S2	4
$\overline{CASEB3}$	3	S1	3
$\overline{CASEB2}$	3	S0	2
$\overline{CASEB1}$	3	$\overline{REFREQ}$	1
$\overline{CASEB0}$	3	$\overline{SWRST}$	2
$\overline{CASOB3}$	3	$\overline{SRASE}$	1
$\overline{CASOB2}$	3		
$\overline{CASOB1}$	3		
$\overline{CASOB0}$	3		
$\overline{FLASHCS}$	2		
$\overline{FLASHRD}$	1		
$\overline{TRIGRST}$	1		
$\overline{XCROE}$	2		
$\overline{XCROIR}$	1		
$\overline{RESET}$	1		
$\overline{WRE}$	1		
$\overline{WRO}$	1		
LED_LAT	1		
$\overline{SELB}$	4		

intel

AP-704

APPLICATION NOTE

A Simple DRAM Controller for 25/16 MHz i960® CA/CF Microprocessors

February 20, 1995

Order Number: 272628-001

A Simple DRAM Controller for 25/16 MHz i960® CA/CF Microprocessors

February 20, 1995

Order Number: 272628-001

1.0 INTRODUCTION

This application note describes a simple DRAM controller for use with 25 and 16 MHz i960® Cx processors. Other application notes are available which describe DRAM controllers for the i960 Cx and Jx processors; see Section 7.0, RELATED INFORMATION for ordering information.

This document contains general DRAM controller theory, state machine definitions and timing diagrams. It also contains the PLD equations used to build and test the prototype design. All timing analysis was verified with Timing Designer*. PLD equations were implemented in ABEL* as a device-independent design. Design features include:

- a simple low-cost non-interleaved design
- an ability to use standard 70 ns DRAM SIMM*
- 2-1-1-1 wait state burst reads
- 2-1-1-1 wait state burst writes

This design was implemented and tested on real hardware. The timing analysis, schematics and PLD files are available through Intel's America's Application Support BBS, at (916) 356-3600.

1.1 Design Goals

The primary goal of this design is to implement a single bank 32-bit DRAM controller with the minimum number of components, using a conventional 72-pin fast page mode DRAM SIMM. Such a design may be useful in small embedded systems where space is at a premium. Accordingly, the controller design avoids such techniques as bank interleaving, write posting, and parity support. The state machines require only a 1x frequency clock - no frequency multipliers or delay elements are needed. As tested, the controller also generates the basic control signals needed to support flash memory, SRAM, and I/O peripherals residing on an 8-bit buffered bus.

Although an Altera* EPX740 FLEXlogic device was used to design the prototype, none of the advanced features of the FLEXlogic family were used, making the design easy to port to other programmable logic families. Appendix B describes a specific implementation in simple PLDs.

1.2 Page Mode DRAM SIMM Review

Page mode DRAM allows faster memory access by keeping the same row address while selecting random column addresses within that row. A new column address is selected by deasserting $\overline{CAS}$ while keeping $\overline{RAS}$ active and then asserting $\overline{CAS}$ with the new column address valid to the DRAM. Page mode operation works very well with burst buses in which a single address cycle can be followed by multiple data cycles.

The individual DRAM $\overline{WE}$ signals are tied to common $\overline{WE}$ pin on the SIMM; this requires the use of early write cycles. In an early write cycle, write data is referenced to the falling edge of $\overline{CAS}$, not the falling edge of $\overline{WE}$.

Each SIMM also has four $\overline{CAS}$ lines, one for every eight (nine) bits in a 32-bit (36-bit) SIMM module. The four $\overline{CAS}$ lines control the writing to individual bytes within each SIMM.

1.3 Burst Capabilities for 32-Bit Bus

A bus access starts by asserting $\overline{ADS}$ in the address cycle, and ends by asserting $\overline{BLAST}$ in the last data cycle. Figure 1 shows $\overline{ADS}$ and $\overline{BLAST}$ timings for a quad-word access.

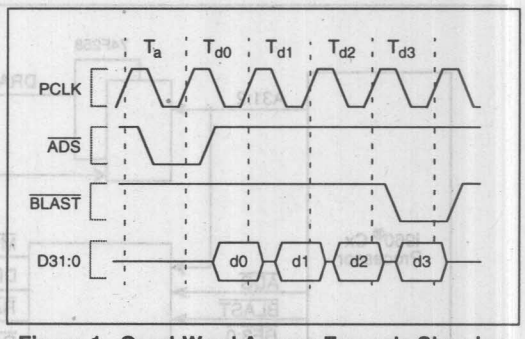


Figure 1. Quad-Word Access Example Showing $\overline{ADS}$ and $\overline{BLAST}$ Timings

2.0 DRAM CONTROLLER OVERVIEW

Figure 2 shows a block diagram of a typical system using this DRAM controller. The design comprises four distinct blocks: control logic, address path, data path, and the DRAM SIMM. This section describes each block.

2.1 Control Logic

The controller consists of several state machines implemented in a programmable logic device (PLD). All state machines are driven by the processor's 1x PCLK.

Design of this controller began by sketching out the desired DRAM control signals on paper, using PCLK as a time base. Individual state machines were then defined by observing the points on the waveforms where the various control signals needed to assert and deassert. These conditions were then recorded as Boolean expressions and transferred to ABEL syntax.

Rather than creating a large state machine to define numerous control outputs, this design uses a large number of simple (two-state) machines — one for each output signal. This technique was chosen to reduce the number of PLD resources required.

The DRAM controller performs address decoding internally. The address bus' upper three bits determine if a processor cycle is to be handled by the DRAM controller. If a finer range is required, additional address bits can be decoded or an external decoder can be added.

Byte and short writes are supported with individual $\overline{CAS}$ signals, one for each byte of the 32-bit memory. Unaccessed bytes during write cycles see a $\overline{RAS}$ only refresh cycle. All write cycles are "early writes", where $\overline{WE}$ is asserted prior to $\overline{CAS}$.

Refresh cycles are generated approximately every 15 µs by dividing down an external 1.8432 MHz baud rate clock. Other refresh clock frequencies can be accommodated with minimal changes. $\overline{CAS}$ before $\overline{RAS}$ refresh cycles are used.

Wait states are inserted by the DRAM controller by providing a $\overline{READY}$ signal to the processor. If additional ready controls are needed for other peripherals, they can be externally OR'ed with the DRAM controller's $\overline{DRDY}$ output.

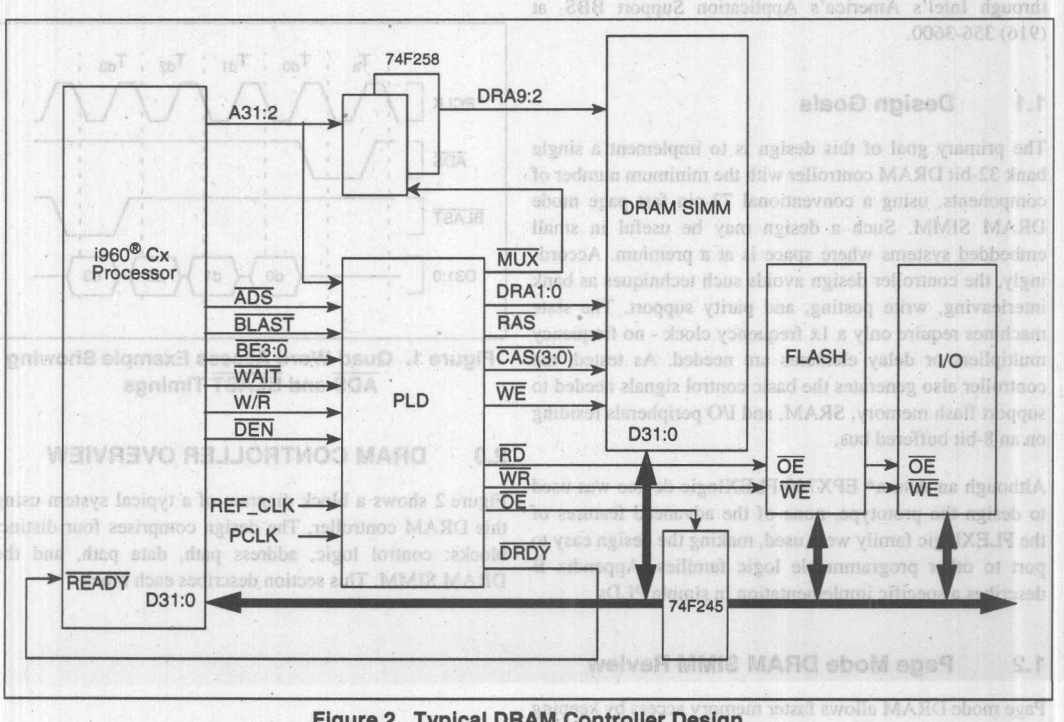


Figure 2. Typical DRAM Controller Design

2.2 Address Path

DRAM address multiplexing is provided by two 74F258 quad 2:1 multiplexers. These devices switch the upper 8 bits of the 10-bit DRAM address bus. The lower two address bits are generated in the PLD to provide faster column address during the page mode bursts. The address multiplexers also provide buffering for the capacitive load of the DRAM SIMM.

Use of larger SIMMs would require an additional 74F258 to switch the extra address lines.

2.3 Data Path

The DRAM data path is unbuffered. All other peripherals in this example are 8-bit devices and are buffered via a 74F245 transceiver. This eliminates the data buffer delay from the DRAM-critical path and places it in the slower (and less critical) I/O path. It also reduces the number of required 74F245 transceivers. Since the DRAM array is the only 32-bit device in the design, the DRAM devices see a very lightly loaded data bus — just the CPU and the 74F245 I/O buffer.

2.4 SIMMS

The SIMM block consists of one standard 72-pin SIMM socket. This design does not use the x36 SIMM parity bits. However, the x36 SIMMs are standard for PCs and workstations; as such, they are readily available. The only penalty is more loading on the address and control lines due to the extra DRAM devices of the x36 SIMM.

All address and control lines use series-damping resistors to control ringing.

3.0 THEORY OF OPERATION

The controller generates the following signals:

- $\overline{RAS}$ — DRAM row address strobe, active low
- $\overline{CAS3:0}$ — Four column address strobes, one for each byte, active low
- $\overline{MUX}$ — Control signal for the 74F258 row/column address multiplexers
- $\overline{DRDY}$ — $\overline{READY}$ control to the i960 Cx processor

The DRAM controller uses the processor's $\overline{READY}$ signal to control wait states. The processor's MCON register is initialized as follows: $N_{XAD} = N_{XDA} = N_{XDD} = 0$, $\overline{READY}$ enable = 1, Burst Enable = 1.

Figure 3 shows non-burst read and write cycles. The DRAM controller begins a processor cycle when it sees $\overline{ADS}$ and a valid address range on the upper address bits A31:29. This occurs on the rising edge of PCLK that ends the address state (T_a) of the bus cycle. On this same edge $\overline{RAS}$ is asserted. One clock later the address multiplexer is switched with the assertion of the $\overline{MUX}$ signal. On the following clock one or more of the $\overline{CAS}$ signals are asserted, based on which byte enables are driven active. $\overline{CAS}$ is low for one clock.

If a burst cycle is requested, $\overline{CAS}$ is deasserted for one clock (hence the one wait state) and reasserted for another clock. This process repeats until $\overline{BLAST}$ is asserted, as shown in Figure 4.

Reads and writes are handled exactly the same way by the RAS and CAS state machines. DRAM $\overline{WE}$ control is provided by the processor's W/$\overline{R}$.

Refresh cycles, as shown in Figure 5, start with the assertion of all four $\overline{CAS}$'s. One clock later $\overline{RAS}$ is asserted and, on the following clock, the $\overline{CAS}$'s are deasserted. On the next clock $\overline{RAS}$ is deasserted. To provide ample row precharge time, a subsequent processor cycle does not start until $\overline{RAS}$ is high for at least two clocks. Refresh requests are recognized during idle bus clocks (T_i) or during the address state (T_a) that starts a processor cycle. Refresh requests are given priority over processor requests. Once a refresh cycle starts, processor requests are delayed until the refresh completes. An example of this can be seen in Figure 8.

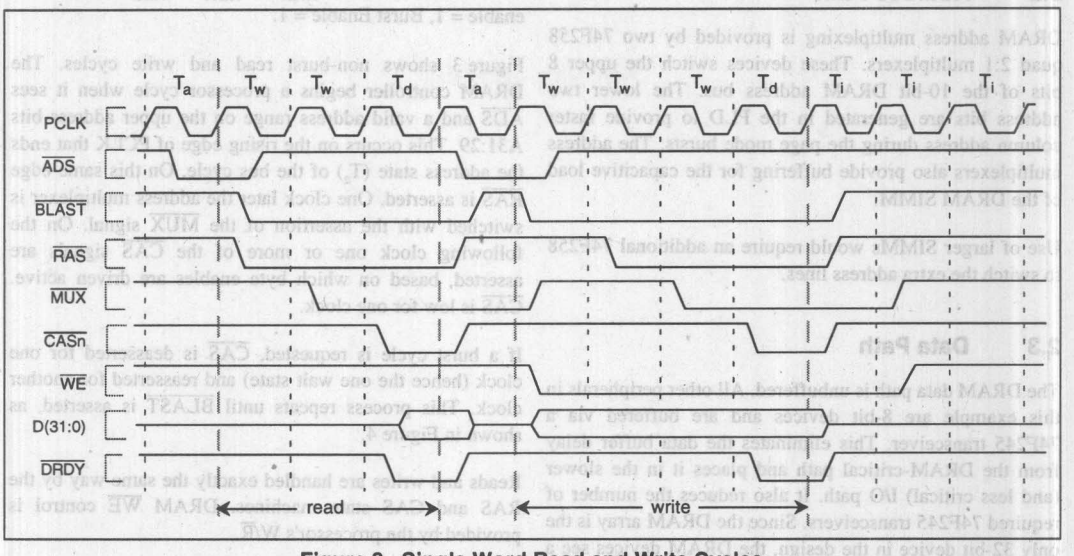


Figure 3. Single Word Read and Write Cycles

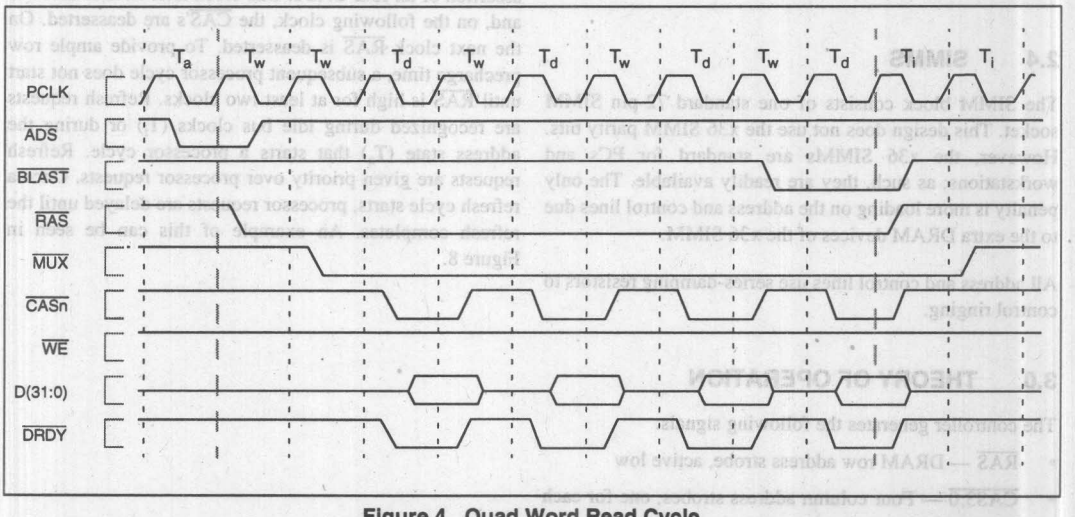


Figure 4. Quad Word Read Cycle

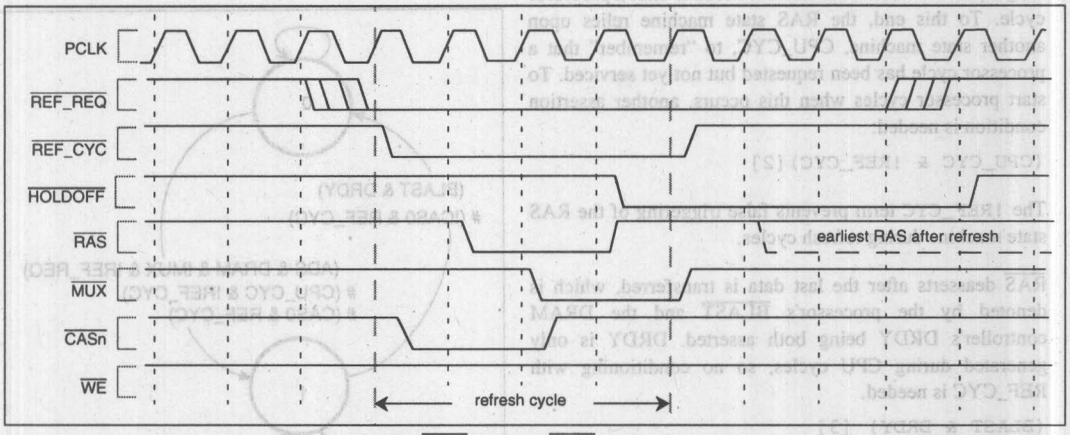


Figure 5. CAS-Before-RAS Refresh Cycle

4.0 STATE MACHINE DESCRIPTIONS

This section details the operation of each state machine that this design uses. All machines are "Mealy" machines, in which the current output is a function only of current state. These machines can be easily implemented in the conventional D flip-flop macrocells in common PLDs.

Most state machines in this design have only two states. The machine transitions from one state to another upon satisfying a Boolean expression; otherwise, it remains in the current state. These expressions consist of:

- **assertion** conditions — the machine transitions from the deasserted (off) state to the asserted (on) state, and
- **deassertion** conditions — the machine transitions from the asserted (on) state to the deasserted (off) state

To illustrate these conditions, several timing diagrams are shown included in this section. Each equation is tagged with a number in square brackets "[]". Use these numbers to identify the corresponding points in the timing diagrams.

All PLD equations are written in ABEL. APPENDIX A, PLD EQUATIONS, contains a listing of the PLD equations file. The state machine transitions described here follow the ABEL conventions for logic operators.

- ! represents NOT, bit-wise negation
- & represents AND
- # represents OR

All signals appearing in logic equations are assumed to be active high. Many of the PLD outputs are active low; when these appear on timing diagrams, the signal names are denoted with an overline (such as $\overline{RAS}$).

4.1 RAS State Machine

The RAS state machine operates in one of two modes: one for CPU cycles and another for refresh cycles. Two other state machine inputs are used:

- REF_REQ state machine — indicates that a refresh request is pending
- REF_CYC state machine — indicates that a refresh cycle is actually underway

The RAS state machine is responsible for recognizing the start of processor cycles. When the DRAM controller is idle (and the RAS precharge time has been met), $\overline{RAS}$ is asserted when $\overline{ADS}$ is asserted and the upper address bits indicate a DRAM address select:

```
(ADS & DRAM & !MUX & !REF_REQ)[1]
```

The `!MUX` term keeps $\overline{RAS}$ from being asserted until $\overline{MUX}$ is deasserted, which provides the minimum 2 PCLK $\overline{RAS}$ precharge time.

The term `!REF_REQ` keeps a processor cycle from being recognized when a refresh request is pending, which gives priority to refresh cycles. However, if the DRAM controller is running a refresh cycle when this occurs, $\overline{ADS}$ is no

longer present when the controller needs to start a processor cycle. To this end, the RAS state machine relies upon another state machine, CPU_CYC, to "remember" that a processor cycle has been requested but not yet serviced. To start processor cycles when this occurs, another assertion condition is needed:

```
(CPU_CYC & !REF_CYC)[2]
```

The `!REF_CYC` term prevents false triggering of the RAS state machine during refresh cycles.

$\overline{RAS}$ deasserts after the last data is transferred, which is denoted by the processor's $\overline{BLAST}$ and the DRAM controller's DRDY being both asserted. DRDY is only generated during CPU cycles, so no conditioning with REF_CYC is needed.

```
(BLAST & DRDY) [3]
```

During refresh, the CAS state machines start the cycle and the RAS machine operates as a slave. For a $\overline{CAS}$-before-$\overline{RAS}$ refresh cycle, $\overline{RAS}$ is asserted one clock after $\overline{CAS}$ is asserted and $\overline{RAS}$ is deasserted one clock after $\overline{CAS}$ is deasserted. Since all four $\overline{CAS}$'s are asserted during refresh, only one (in this case $\overline{CAS0}$) is needed to trigger the RAS state machine. The assertion condition during refresh cycles is:

```
(CAS0 & REF_CYC) [4]
```

Deassertion is:

```
(!CAS0 & REF_CYC) [5]
```

At this point the two operating modes of the RAS machine are combined and a single set of assertion/deassertion equations can be written. Assertion conditions are:

```
  (ADS & DRAM & !MUX & !REF_REQ) [1]
# (CPU_CYC & !REF_CYC) [2]
# (CAS0 & REF_CYC) [4]
```

Deassertion conditions are:

```
  (BLAST & DRDY) [3]
# (!CAS0 & REF_CYC) [5]
```

The following figures illustrate these conditions superimposed on timing diagrams. Figure 7 shows the added wait state needed for RAS precharge when two CPU cycles occur back-to-back. Figure 8 shows what happens when a refresh cycle preempts a CPU cycle. In this case four additional wait states are inserted. Figure 9 shows refresh cycles.

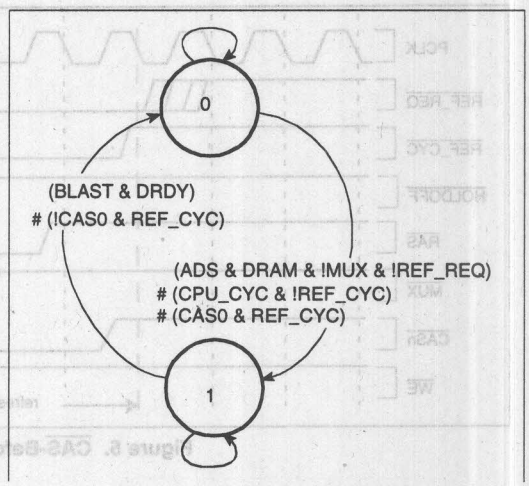


Figure 6. RAS State Machine

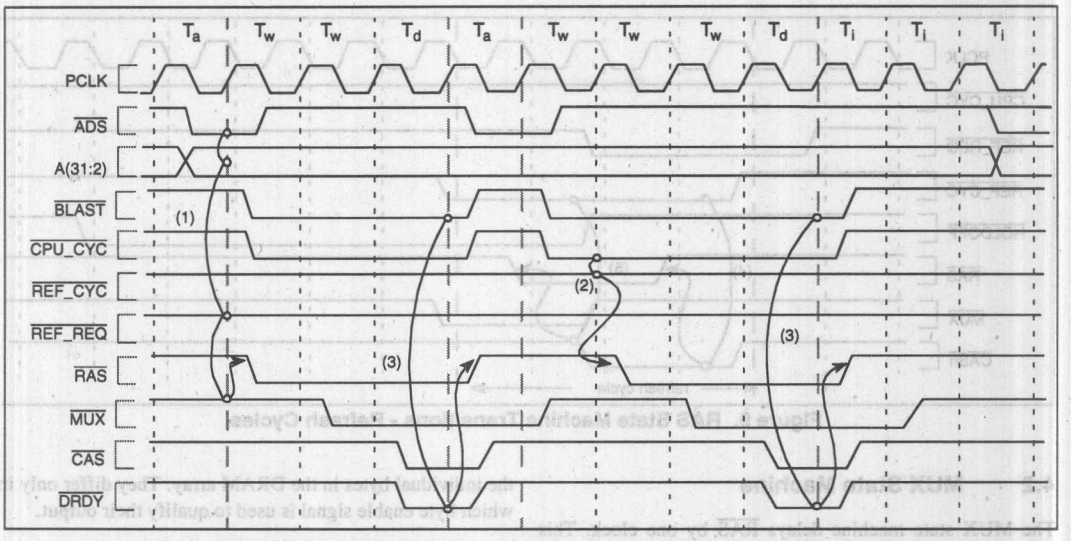


Figure 7. RAS State Machine Transitions - Back-to-Back Processor Cycles

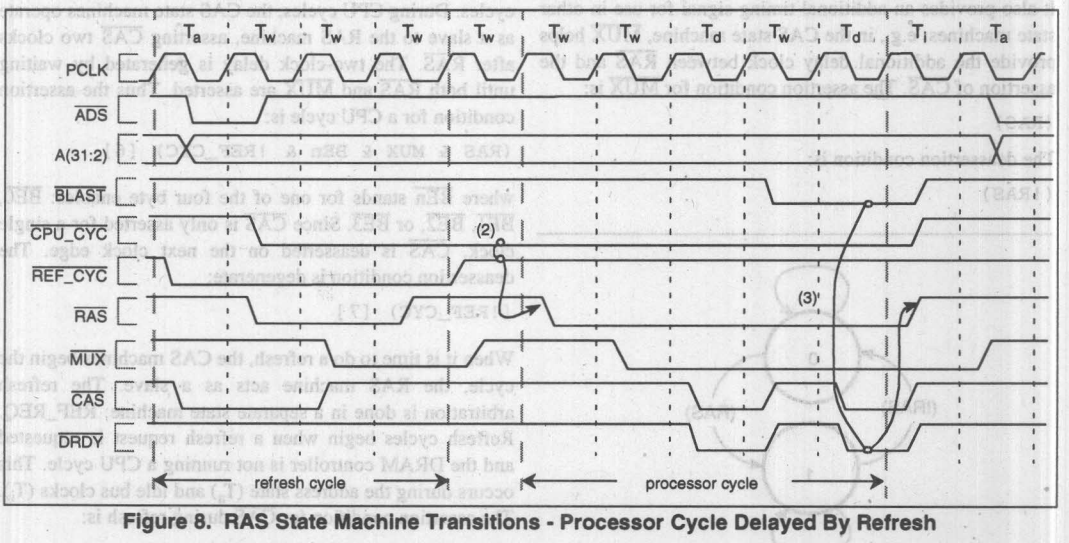


Figure 8. RAS State Machine Transitions - Processor Cycle Delayed By Refresh

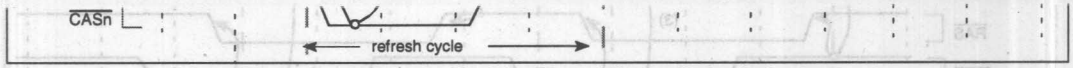


Figure 9. RAS State Machine Transitions - Refresh Cycles

4.2 MUX State Machine

The MUX state machine delays $\overline{RAS}$ by one clock. This switches the address multiplexer from "row" to "column" in time for the column address to stabilize before $\overline{CAS}$ asserts. It also provides an additional timing signal for use in other state machines; e.g., in the CAS state machine, $\overline{MUX}$ helps provide the additional delay clock between $\overline{RAS}$ and the assertion of $\overline{CAS}$. The assertion condition for $\overline{MUX}$ is:

```
(RAS)
```

The deassertion condition is:

```
(!RAS)
```

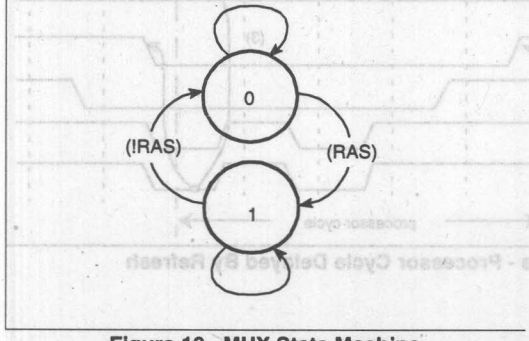


Figure 10. MUX State Machine

4.3 CAS State Machines

Four nearly-identical CAS state machines (CAS0, CAS1, CAS2, and CAS3) provide the column address strobes for the individual bytes in the DRAM array. They differ only in which byte enable signal is used to qualify their output.

Like the RAS state machine, the CAS state machines operate in one of two modes: CPU cycles and refresh cycles. During CPU cycles, the CAS state machines operate as a slave to the RAS machine, asserting $\overline{CAS}$ two clocks after $\overline{RAS}$. The two-clock delay is generated by waiting until both $\overline{RAS}$ and $\overline{MUX}$ are asserted. Thus the assertion condition for a CPU cycle is:

```
(RAS & MUX & BEn & !REF_CYC) [6]
```

where $\overline{BEn}$ stands for one of the four byte enables: $\overline{BE0}$, $\overline{BE1}$, $\overline{BE2}$, or $\overline{BE3}$. Since $\overline{CAS}$ is only asserted for a single clock, $\overline{CAS}$ is deasserted on the next clock edge. The deassertion condition is degenerate:

```
(!REF_CYC) [7]
```

When it is time to do a refresh, the CAS machines begin the cycle, the RAS machine acts as a slave. The refresh arbitration is done in a separate state machine: REF_REQ. Refresh cycles begin when a refresh request is requested and the DRAM controller is not running a CPU cycle. This occurs during the address state (T_a) and idle bus clocks (T_i). The assertion condition for $\overline{CAS}$ during refresh is:

```
(REF_REQ & !CPU_CYC) [8]
```

The deassertion condition is (see (4) in Figure 13):

```
(RAS & REF_CYC) [9]
```

$\overline{CAS}$ is deasserted one clock after $\overline{RAS}$ goes low.

The CPU refresh modes can now be combined into a single set to generate the state diagram shown in Figure 11. Assertion conditions are:

```
(RAS & MUX & BEn & !REF_CYC) [6]
# (REF_REQ & !CPU_CYC) [8]
```

Deassertion:

```
(!REF_CYC) [7]
# (RAS & REF_CYC) [9]
```

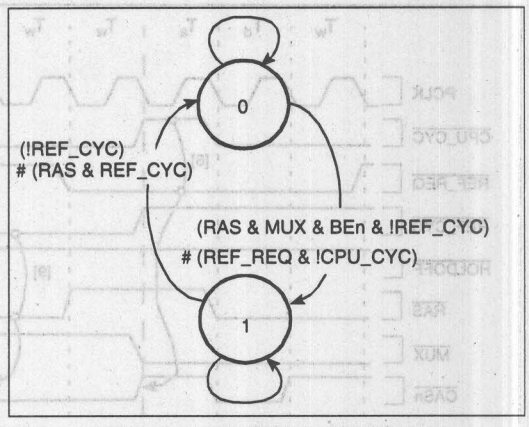


Figure 11. CAS State Machines

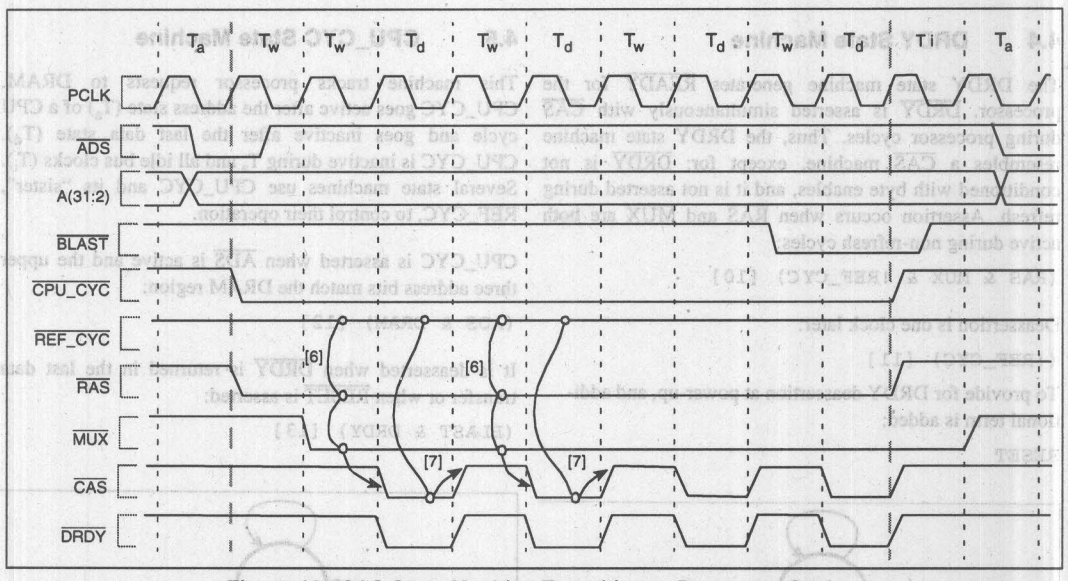


Figure 12. CAS State Machine Transitions - Processor Cycle

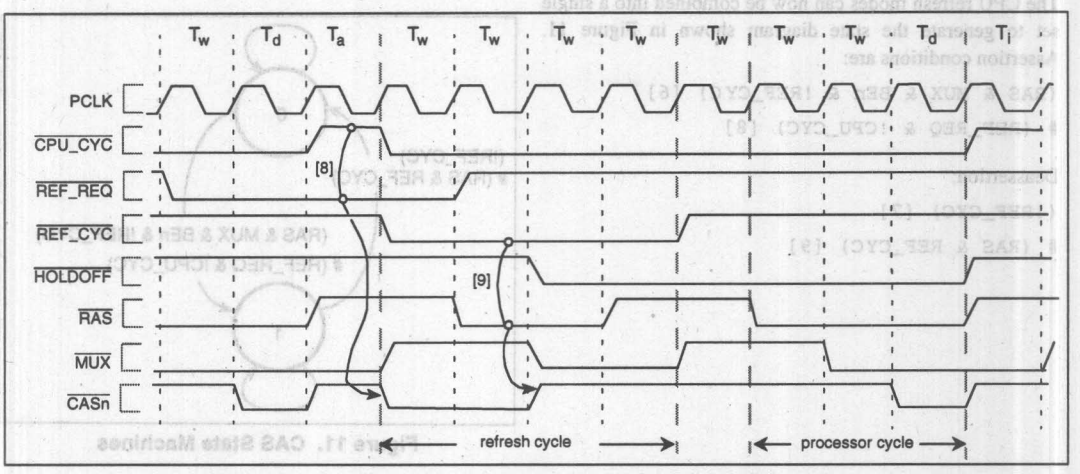


Figure 13. CAS State Machine Transitions - Refresh Cycles

4.4 DRDY State Machine

The DRDY state machine generates $\overline{\text{READY}}$ for the processor. $\overline{\text{DRDY}}$ is asserted simultaneously with $\overline{\text{CAS}}$ during processor cycles. Thus, the DRDY state machine resembles a $\overline{\text{CAS}}$ machine, except for: $\overline{\text{DRDY}}$ is not conditioned with byte enables, and it is not asserted during refresh. Assertion occurs when $\overline{\text{RAS}}$ and $\overline{\text{MUX}}$ are both active during non-refresh cycles:

```
(RAS & MUX & !REF_CYC) [10]
```

Deassertion is one clock later:

```
(!REF_CYC) [11]
```

To provide for DRDY deassertion at power-up, and additional term is added:

```
RESET
```

4.5 CPU_CYC State Machine

This machine tracks processor requests to DRAM. CPU_CYC goes active after the address state (T_a) of a CPU cycle and goes inactive after the last data state (T_d). CPU_CYC is inactive during T_a and all idle bus clocks (T_i). Several state machines use CPU_CYC and its "sister", REF_CYC, to control their operation.

CPU_CYC is asserted when $\overline{\text{ADS}}$ is active and the upper three address bits match the DRAM region:

```
(ADS & DRAM) [12]
```

It is deasserted when $\overline{\text{DRDY}}$ is returned in the last data transfer or when $\overline{\text{RESET}}$ is asserted:

```
(BLAST & DRDY) [13]
```

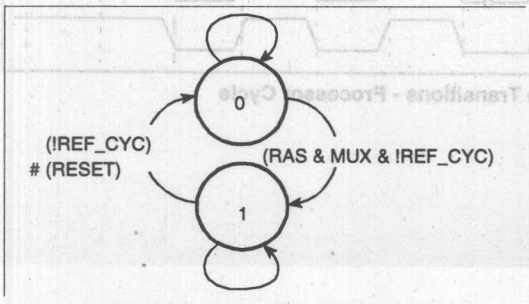


Figure 14. DRDY State Machine

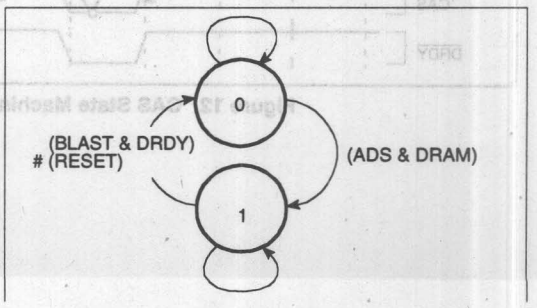


Figure 15. CPU_CYC State Machine

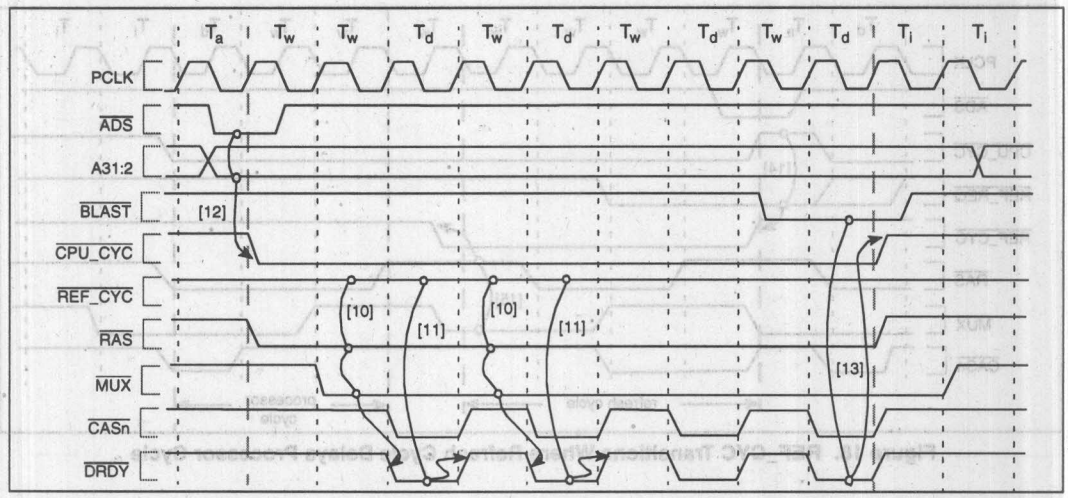


Figure 16. CPU_CYC and DRDY State Machine Transitions

4.6 REF_CYC State Machine

The REF_CYC State Machine indicates when the DRAM controller is performing a refresh cycle. Refresh cycles are requested by another state machine, the REF_REQ state machine, every 15 µs. Refresh cycles may begin during any idle bus clock (T_i) or during the address state (T_a) that starts a processor bus cycle. If a refresh request occurs during a processor cycle, $\overline{\text{REF\_CYC}}$ is not asserted until the next T_i or T_a state.

An idle bus and the address state look the same to the REF_CYC state machine. Both are signified by having $\overline{\text{CPU\_CYC}}$ deasserted.

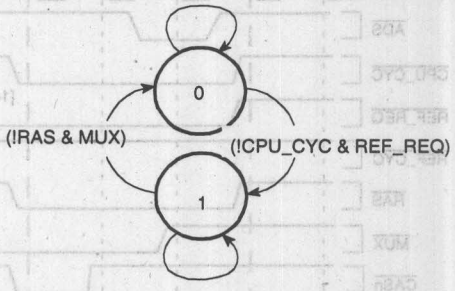


Figure 17. REF_CYC State Machine

The assertion conditions for $\overline{\text{REF\_CYC}}$ are:

```
(!CPU_CYCLE & REF_REQ) [14]
```

REF_CYC is not deasserted until $\overline{\text{RAS}}$ has returned high after the refresh cycle. Since $\overline{\text{RAS}}$ is high during the first clock of a refresh cycle, the deassertion condition includes $\overline{\text{MUX}}$ to keep $\overline{\text{REF\_CYC}}$ from going prematurely inactive. The deassertion conditions are:

```
(!RAS & MUX) [15]
```

Figure 18 shows the worst case delay of a processor cycle due to a DRAM refresh. Figure 19 shows the worst case delay of a refresh cycle due to a processor cycle.

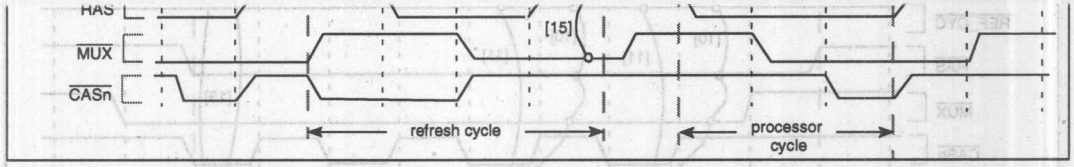


Figure 18. REF_CYC Transitions Where Refresh Cycle Delays Processor Cycle

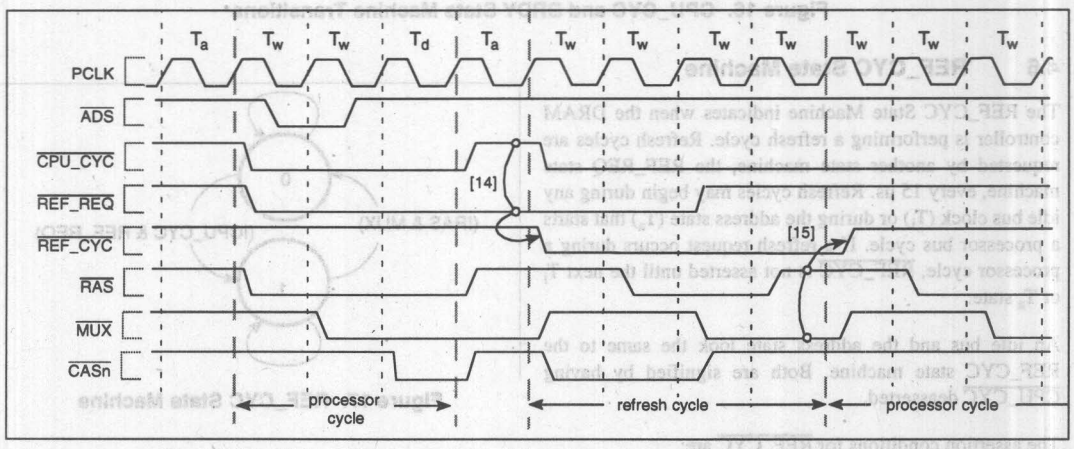


Figure 19. REF_CYC Transitions Where Processor Cycle Delays Refresh

4.7 REF_REQ State Machine

The REF_REQ state machine is an edge-detected version of the refresh counter's most significant bit. It is also synchronized to PCLK. $\overline{\text{REF\_REQ}}$ is asserted as soon as the counter ($\overline{\text{R4}}$) is set and $\overline{\text{HOLDOFF}}$ is deasserted. $\overline{\text{HOLDOFF}}$, as shown below, prevents $\overline{\text{REF\_REQ}}$ from being reasserted until $\overline{\text{R4}}$ changes state. $\overline{\text{REF\_REQ}}$ is deasserted as soon as a refresh cycle is actually started. Assertion conditions are:

```
(R4 & !HOLDOFF) [16]
```

Deassertion conditions are:

```
(REF_CYC) [17]
```

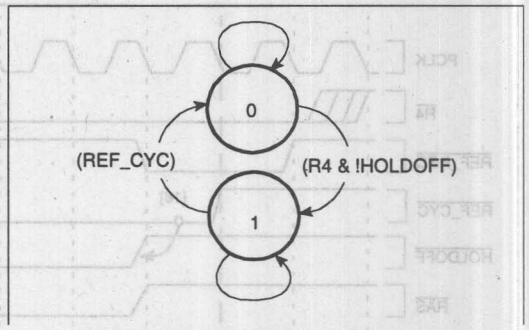


Figure 20. REF_REQ State Machine

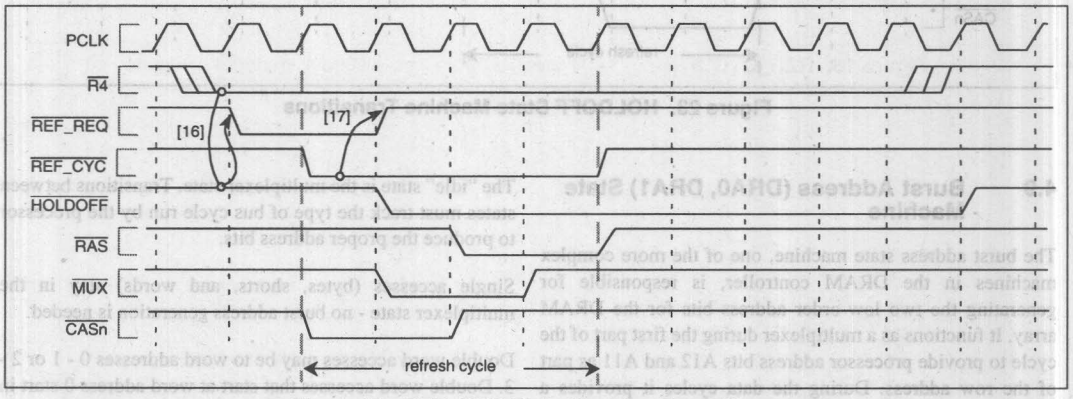


Figure 21. REF_REQ State Machine Transitions

4.8 HOLDOFF State Machine

As noted in Section 4.7, the HOLDOFF state machine provides an edge-detect function for the $\overline{\text{REF\_REQ}}$ input. $\overline{\text{HOLDOFF}}$ is asserted during a refresh cycle at the point when RAS is asserted. $\overline{\text{HOLDOFF}}$ assertion prevents another refresh cycle from being requested by the REF_REQ state machine until the refresh timer $\overline{\text{R4}}$ is deasserted. Assertion conditions are:

```
(REF_CYC) [18]
```

Deassertion conditions are:

```
(!R4) [19]
```

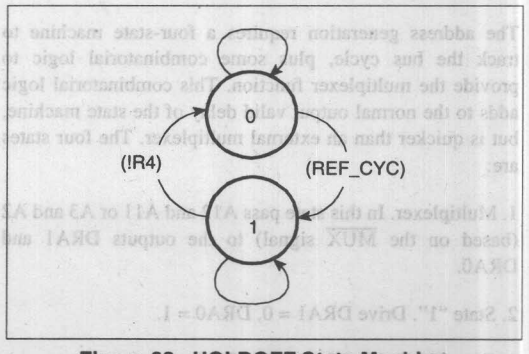


Figure 22. HOLDOFF State Machine

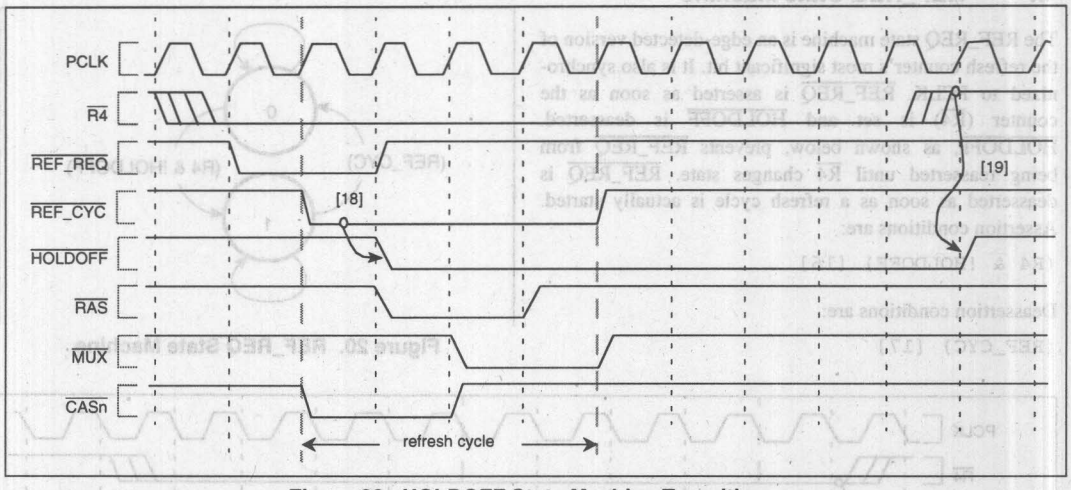


Figure 23. HOLDOFF State Machine Transitions

4.9 Burst Address (DRA0, DRA1) State Machine

The burst address state machine, one of the more complex machines in the DRAM controller, is responsible for generating the two low-order address bits for the DRAM array. It functions as a multiplexer during the first part of the cycle to provide processor address bits A12 and A11 as part of the row address. During the data cycles it provides a synthesized version of processor bits A3 and A2. Generating A3 and A2 in the state machine reduces the delay in the critical path from column address to data out.

The address generation requires a four-state machine to track the bus cycle, plus some combinatorial logic to provide the multiplexer function. This combinatorial logic adds to the normal output valid delay of the state machine, but is quicker than an external multiplexer. The four states are:

1. Multiplexer. In this state pass A12 and A11 or A3 and A2 (based on the $\overline{MUX}$ signal) to the outputs DRA1 and DRA0.

2. State "1". Drive DRA1 = 0, DRA0 = 1.

3. State "2". Drive DRA1 = 1, DRA0 = 0.

4. State "3": Drive DRA1 = 1, DRA0 = 1.

The "idle" state is the multiplexer state. Transitions between states must track the type of bus cycle run by the processor to produce the proper address bits.

Single accesses (bytes, shorts, and words) stay in the multiplexer state - no burst address generation is needed.

Double word accesses may be to word addresses 0 - 1 or 2 - 3. Double word accesses that start at word address 0 start in the multiplex state (to allow address 0 to flow through), move to state "1" to provide word address 1, then return to the multiplexer state (idle). Accesses that start at word address 2 jump from the multiplexer state to state "3" before returning to idle.

Triple word accesses, by definition, always start with word address 0. They transition from multiplexer state, state "1", state "2", and then back to idle.

Quad word accesses always begin with word address 0 and cycle though all four states in order.

The machine stays in the multiplexer state until a multi-word bus cycle is detected ($\overline{DRDY}$ asserted while $\overline{BLAST}$ is deasserted). Based on the state of A3 and A2 during the first access, it advances to either state "1" or state "3". While in state "1" through "3" it returns to the multiplexer state when the bus cycle ends ($\overline{DRDY}$ and $\overline{BLAST}$ asserted).

The state machine is defined in a "present state - next state" syntax as follows:

```
STATE_MUX:
  !A3 & !A2 & DRDY & !BLAST -> STATE_1
  A3 & !A2 & DRDY & !BLAST -> STATE_3
  else      -> STATE_MUX

STATE_1:
  DRDY & !BLAST -> STATE_2
       DRDY & BLAST -> STATE_MUX
  else      -> STATE_1

STATE_2:
  DRDY & !BLAST -> STATE_3
  DRDY & BLAST -> STATE_MUX
  else      -> STATE_2

STATE_3:
  DRDY & BLAST -> STATE_MUX
  else      -> STATE_3
```

While in the MUX state, DRA1 and DRA0 are either the row address (A12 and A11) or the column address (A3 and A2) based on the $\overline{MUX}$ signal. When in states 1 through 3, DRA1 and DRA0 are the binary value of those states, as follows:

```
DRA1 = (STATE_MUX & !MUX & A12) #
(STATE_MUX & MUX & A3) #
          STATE_2 # STATE_3

DRA0  = (STATE_MUX & !MUX & A11) #
(STATE_MUX & MUX & A2) #
         STATE_1 # STATE_3
```

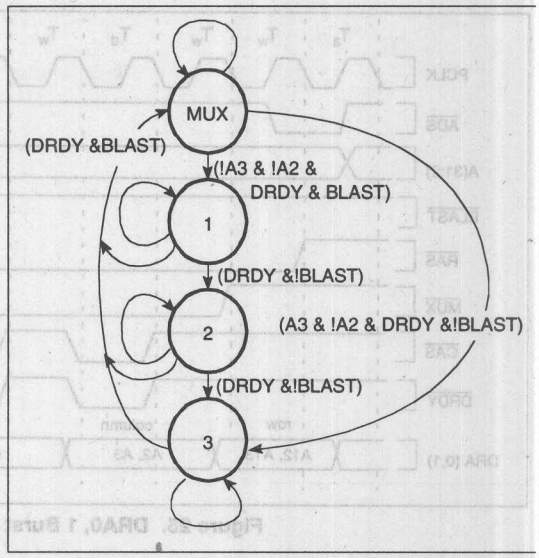


Figure 24. DRA0, DRA1 State Machine

4.10 Refresh Divider State Machine

The 1.8432 MHz baud rate clock is divided by 28 to produce a $\overline{REF\_REQ}$ with a 15 µs period. This is performed by a five-bit binary up counter. $\overline{R4}$ is the most significant bit of the counter. See APPENDIX A for details. Use of a different refresh clock frequency would require adjusting the division ratio to maintain the 15 µs period.

This state machine may be replaced with an external refresh timer. In this case, an asynchronous square wave with a period of 15 µs may be substituted for the $\overline{R4}$ input to the REF_REQ and HOLDOFF state machines.

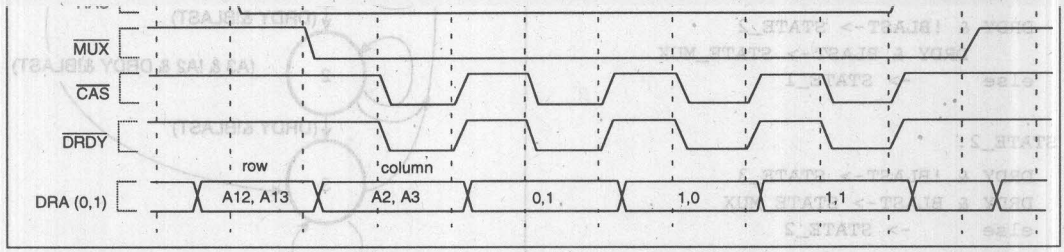


Figure 25. DRA0, 1 Burst Address State Transitions

4.11 Read Strobe State Machine

This state machine produces a general-purpose read strobe ($\overline{RD}$) for FLASH memory, SRAM, and other I/O peripherals. $\overline{RD}$ is asserted at the start of all read cycles without any address qualification. $\overline{RD}$ is deasserted when $\overline{BLAST}$ is issued by the processor. It is assumed that the internal i960 Cx processor's wait state generator is used to control all such bus cycles, since no $\overline{READY}$ is generated by this logic.

$\overline{RD}$ assertion conditions are:

```
(ADS & !W/R) [20]
```

Deassertion conditions are:

```
(BLAST) [21]
```

Additionally, $\overline{RD}$ is deasserted upon power-up by the $\overline{RESET}$ signal.

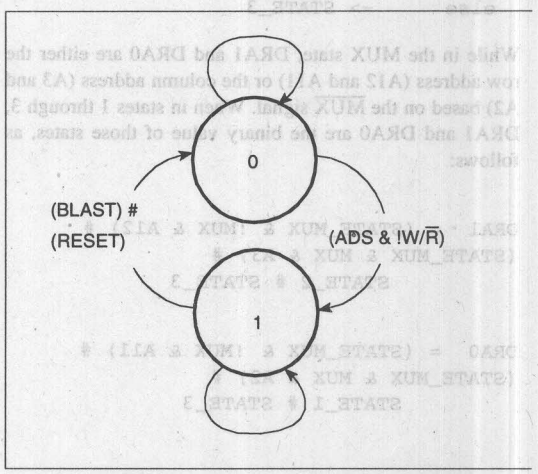


Figure 26. RD (Read Strobe) State Machine

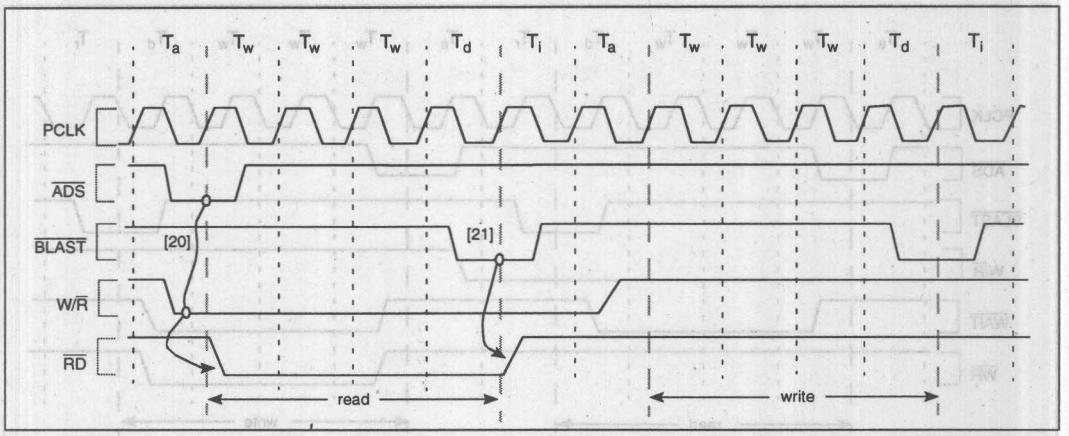


Figure 27. RD State Machine Transitions

5.0 COMBINATORIAL OUTPUTS

The few remaining general-purpose controls can be generated from processor signals with simple combinatorial logic. Among these are the SRAM and I/O write strobe ($\overline{WR}$), the DRAM write enable ($\overline{WE}$), and the transceiver output enable ($\overline{245OE}$).

5.1 I/O Write Strobe

For regions controlled by the internal wait state generator with more than one wait state, the processor's $\overline{WAIT}$ signal can be used to generate a suitable write strobe for I/O peripherals (see Figure 28):

```
WR# = (W/R & WAIT)
```

5.2 DRAM Write Enable

An inverted version of the processor's W/$\overline{R}$ signal can be used for this purpose with one modification. Most DRAMs require that $\overline{WE}$ is high (deasserted) during refresh cycles. Since refresh cycles may occur while the processor is waiting to perform a DRAM access, the state of the W/$\overline{R}$ signal cannot be guaranteed. Accordingly W/$\overline{R}$ needs to be gated with $\overline{REF\_CYC}$ as follows:

```
WE# = (W/R & !REF_CYC)
```

5.3 Transceiver Control

In the tested design, an I/O transceiver isolates the 8-bit I/O bus from the processor and DRAM array. The 74F245 device requires two controls: a direction select and an output enable. To eliminate bus contention, the direction should be changed only when the output enable is deasserted.

Direction select is taken directly from the processor's DT/$\overline{R}$ control. Output enable is generated by a combination of the processor's $\overline{DEN}$ signal and an address decode of the upper address bits. This defines a section of memory that activates the transceiver, keeping it from interfering with DRAM cycles.

```
OE245# = (DEN & IO_DECODE)
```

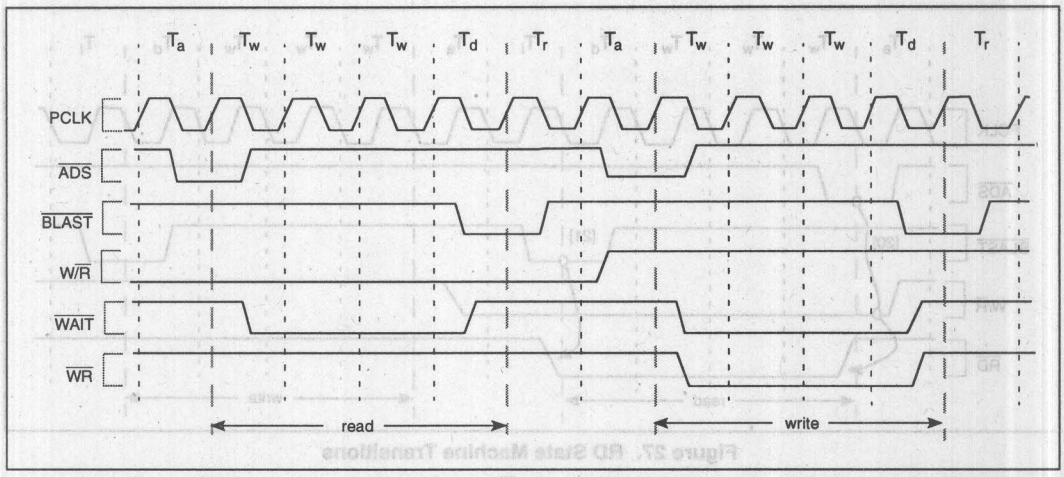


Figure 28. W/$\overline{R}$ Strobe Generation

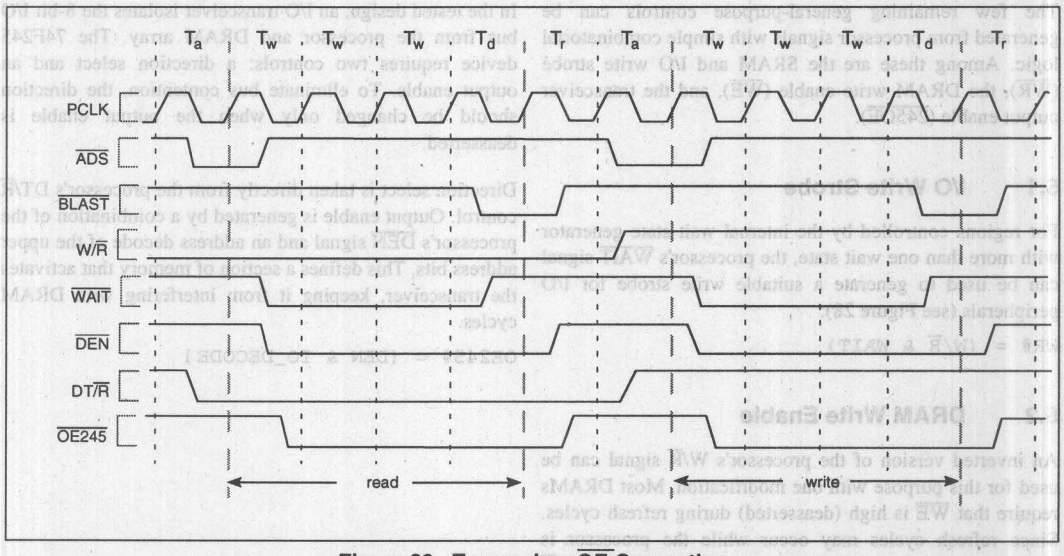


Figure 29. Transceiver $\overline{OE}$ Generation

6.0 CONCLUSION

In conclusion, this application note describes a simple DRAM controller for use with i960 CA/CF 16/25 MHz processors. This DRAM controller was built and tested for validation purposes. The PLD equations used to build and test the prototype design were created in ABEL. All timing analysis was verified with Timing Designer. Schematics were created with OrCAD. The timing analysis, schematics and PLD files are available through Intel's America's Application Support BBS.

7.0 RELATED INFORMATION

This application note is one of four that are related to DRAM controllers for the i960 processors. The following table shows the documents and order numbers:

Document Name	App. Note #	Order #
DRAM Controller for the 40 MHz i960® CF Microprocessors	AP-706	272655
DRAM Controller for the i960® Jx Microprocessors	AP-712	272674
DRAM Controller for 33 MHz i960® CA/CF Microprocessors	AP-703	272627

To receive these documents or any other available Intel literature, contact:

Intel Corporation
Literature Sales
P.O. Box 7641
Mt. Prospect IL 60056-7641
1-800-879-4683

To receive files that contain the timing analysis, schematics and PLD equations for this and the other DRAM controller application notes, contact:

Intel Corporation
America's Application Support BBS
916-356-3600

7.0 RELATED INFORMATION

This application note is one of four that are related to DRAM controllers for the i960 processors. The following table shows the documents and order numbers:

Document Name	App. Note #	Order #
DRAM Controller for the 40 MHz i960® CF Microprocessors	AP-708	272658
DRAM Controller for the i960® Jx Microprocessors	AP-712	272574
DRAM Controller for 33 MHz i960® CA/CF Microprocessors	AP-703	272627

To receive these documents or any other available Intel literature, contact:

Intel Corporation
Literature Sales
P.O. Box 7641
Mt. Prospect IL 60056-7641
1-800-879-4683

To receive files that contain the timing analysis, schematics and PLD equations for this and the other DRAM controller application notes, contact:

Intel Corporation
America's Application Support BBS
916-356-3600

APPENDIX A
PLD EQUATIONS

Table A-1 contains the PLD equations which were used to build and test the prototype design. Table A-1 defines signal and product term allocation. The PLD equations were created in ABEL as a device-independent design. Using the ABEL* software, a PDS file was created and subsequently imported into PLDSHELL* software. PLDSHELL was used to fit the design into an Altera EPX780 FLEXlogic* PLD. PLDSHELL was also utilized to create the JEDEC file, and to simulate the design.

In addition, this appendix contains a table listing the number of product terms used by each macrocell.

Table A-1. 33 MHz Simple DRAM Controller PLD Equations (Sheet 1 of 9)

```
module  DRAM
title   '960CA DRAM Controller (21121)
Rick Schue  Intel Corp.  01/25/95'

    H,L,Ck,X = 1, 0, .C., .X.;

" Device independent design

" input pin assignments

    PCLK            pin;                " 25.000 MHz clock for state variables
    REFCLK          pin;                " 1.8432 MHz clock for refresh timer
    !ADS            pin;                " CPU ADS# control
    !BLAST          pin;                " CPU BLAST# control
    !RESET          pin;                " system reset (active low)
    A31             pin;                " CPU address lines
    A30             pin;                "
    A29             pin;                "
    A12             pin;                " address lines used to generate
    A11             pin;                " DRA1,0
    A3              pin;                "
    A2              pin;                "
    !BE3            pin;                " CPU byte enables
    !BE2            pin;                "
    !BE1            pin;                "
    !BE0            pin;                "
    !READ           pin;                " CPU W!R# control
    !WAIT           pin;                " CPU WAIT# control
    !DEN            pin;                " CPU DEN# control

" output pin assignments
    !RAS            pin  ISTYPE 'reg';  " DRAM RAS
    !CAS3           pin  ISTYPE 'reg';  " DRAM CAS
    !CAS2           pin  ISTYPE 'reg';  " DRAM CAS
    !CAS1           pin  ISTYPE 'reg';  " DRAM CAS
    !CAS0           pin  ISTYPE 'reg';  " DRAM CAS
    DRA1            pin  ISTYPE 'com';  " DRAM address outputs (combinatorial)
    DRA0            pin  ISTYPE 'com';  " DRAM address outputs (combinatorial)
    !DRDY           pin  ISTYPE 'reg';  " DRAM ready
    !MUX            pin  ISTYPE 'reg';  " row/column control: 1=row 0=col
    !WE             pin  ISTYPE 'com';  " DRAM WE#
```

Table A-1. 33 MHz Simple DRAM Controller PLD Equations (Sheet 2 of 9)

```
    !RD             pin  ISTYPE 'reg';  " I/O READ
    !WR             pin  ISTYPE 'com';  " I/O WRITE (combinatorial)
    !OE245          pin  ISTYPE 'com';  " 74F245 transceiver control for I!O bus

" internal state variables - these do not have to be assigned to pins unless
" one wishes to use them for debug.
    !CPU_CYC        node ISTYPE 'reg';  " CPU cycle pending
    !REF_REQ        node ISTYPE 'reg';  " refresh request (active low)
    !REF_CYC        node ISTYPE 'reg';  " Refresh cycle pending
    !HOLDOFF        node ISTYPE 'reg';  " Refresh holdoff
    S1,S0           node ISTYPE 'reg';  " DRA1,0 state bits
    R0,R1,R2,R3,R4  node ISTYPE 'reg';  " Refresh timer (divide by 28)

" region decode:
"  to conserve pins in this simple example, addresses are decoded into only
"  eight regions.

    MEMADDR = [A31,A30,A29,X, X,X,X,X, X,X,X,X, X,X,X,X,
                      X,X,X,X, X,X,X,X, X,X,X,X, X,X,X,X];

" state machine vector definitions
    RAS_MACHINE     = [RAS];
    MUX_MACHINE     = [MUX];
    CAS0_MACHINE    = [CAS0];
    CAS1_MACHINE    = [CAS1];
    CAS2_MACHINE    = [CAS2];
    CAS3_MACHINE    = [CAS3];
    DRDY_MACHINE    = [DRDY];
    RD_MACHINE      = [RD];
    CPU_CYC_MACHINE = [CPU_CYC];
    REF_REQ_MACHINE = [REF_REQ];
    REF_CYC_MACHINE = [REF_CYC];
    HOLDOFF_MACHINE = [HOLDOFF];
    BURST_MACHINE   = [S1,S0];
    REFRESH         = [R4,R3,R2,R1,R0];

" literals
    IDLE      = ^b0;
    ACTIVE    = ^b1;
    STATE_MUX = ^b00;
    STATE_1   = ^b01;
    STATE_2   = ^b10;
    STATE_3   = ^b11;

" modify the following address assignments to relocate IO and/or DRAM into
" other memory regions.  In this example:
"          DRAM  0xc0000000 to 0xdfffffff
"          I/O   0x20000000 to 0x3fffffff (UART)
"                0x60000000 to 0x7fffffff (parallel port)
"                0xe0000000 to 0xffffffff (FLASH boot ROM)

    DRAM  = (MEMADDR >= ^hC0000000) & (MEMADDR <= ^hDFFFFFFF);
    IO    = ((MEMADDR >= ^h20000000) & (MEMADDR <= ^h3FFFFFFF)) #
            ((MEMADDR >= ^h60000000) & (MEMADDR <= ^h7FFFFFFF)) #
            ((MEMADDR >= ^hE0000000) & (MEMADDR <= ^hFFFFFFFF));
```

Table A-1. 33 MHz Simple DRAM Controller PLD Equations (Sheet 3 of 9)

```
equations

" DRAM burst address multiplexer
"   state MUX   DRA1   DRA0
"    00    0     A12    A11      pass through ROW address
"    00    1      A3     A2      pass through COL address
"    01    x      0      1       burst address
"    10    x      1      0       burst address
"    11    x      1      1       burst address

    DRA1  = (!S1 & !S0 & !MUX & A12) #
            (!S1 & !S0 & MUX & A3) # (S1 & !S0) # (S1 & S0);
    DRA0  = (!S1 & !S0 & !MUX & A11) #
            (!S1 & !S0 & MUX & A2) # (!S1 & S0) # (S1 & S0);

" simple I/O control signals
    WE    = (!READ & !REF_CYC);
    WR    = (WAIT & !READ);
    OE245 = (DEN & IO);

" the following state machines are synchronous with the CPU
    RAS.clk         = PCLK;
    MUX.clk         = PCLK;
    CAS0.clk        = PCLK;
    CAS1.clk        = PCLK;
    CAS2.clk        = PCLK;
    CAS3.clk        = PCLK;
    DRDY.clk        = PCLK;
    RD.clk          = PCLK;
    CPU_CYC.clk     = PCLK;
    REF_REQ.clk     = PCLK;
    REF_CYC.clk     = PCLK;
    HOLDOFF.clk     = PCLK;
    [S1,S0].clk     = PCLK;

" the refresh divider runs off a second (asynchronous) clock
" if a higher frequency REFCLK is used, more bits may be required here
   [R4,R3,R2,R1,R0].clk = REFCLK;
```

Table A-1. 33 MHz Simple DRAM Controller PLD Equations (Sheet 4 of 9)

```
"-------------------------------------------------------------------------------

state_diagram BURST_MACHINE

" This state machine controls the DRAM A1,A0 multiplexer described
" above in the EQUATIONS section.

    state STATE_MUX:
        if      (!A3 & !A2 & DRDY & !BLAST)   then STATE_1;
        else if ( A3 & !A2 & DRDY & !BLAST)   then STATE_3;
        else    STATE_MUX;

    state STATE_1:
        if      (DRDY & !BLAST)    then STATE_2;
        else if (DRDY &  BLAST)    then STATE_MUX;
        else    STATE_1;

    state STATE_2:
        if      (DRDY & !BLAST)    then STATE_3;
        else if (DRDY &  BLAST)    then STATE_MUX;
        else    STATE_2;

    state STATE_3:
        if      (DRDY &  BLAST)    then STATE_MUX;
        else    STATE_3;

"-------------------------------------------------------------------------------

state_diagram CPU_CYC_MACHINE

" This machine tracks all DRAM cycles requested by the CPU

    state IDLE:
        if (ADS & DRAM) then ACTIVE;
        else IDLE;

    state ACTIVE:
        if (BLAST & DRDY # RESET) then IDLE;
        else ACTIVE;
```

```
    state ACTIVE:
        if (!RAS) then IDLE;
        else ACTIVE;

"-------------------------------------------------------------------------

state_diagram RAS_MACHINE

" This generates RAS for a DRAM bank.  A single RAS controls all bytes in
" that bank.  If additional banks are desired, extra RAS's will have to be
" generated with similar state machines.

    state IDLE:
        if (ADS & DRAM & !MUX & !REF_REQ) #
           (CPU_CYC & !REF_CYC) #
           (REF_CYC & CAS0) then ACTIVE;
        else IDLE;

    state ACTIVE:
        if (BLAST & DRDY) # (REF_CYC & !CAS0) then IDLE;
        else ACTIVE;

"-------------------------------------------------------------------------

state_diagram CAS0_MACHINE

" The following machines generate individual CAS's for each byte of the
" DRAM array.  Their only differences are the byte enables that enable them

    state IDLE:
        if (RAS & BE0 & !REF_CYC & MUX) # (REF_REQ & !CPU_CYC) then ACTIVE;
        else IDLE;

    state ACTIVE:
        if (!REF_CYC) # (REF_CYC & RAS) then IDLE;
        else ACTIVE;
```

```
state_diagram CAS2_MACHINE

    state IDLE:
        if (RAS & BE2 & !REF_CYC & MUX) # (REF_REQ & !CPU_CYC) then ACTIVE;
        else IDLE;
    state ACTIVE:
        if (!REF_CYC) # (REF_CYC & RAS) then IDLE;
        else ACTIVE;

"-------------------------------------------------------------------------

state_diagram CAS3_MACHINE

    state IDLE:
        if (RAS & BE3 & !REF_CYC & MUX) # (REF_REQ & !CPU_CYC) then ACTIVE;
        else IDLE;

    state ACTIVE:
        if (!REF_CYC) # (REF_CYC & RAS) then IDLE;
        else ACTIVE;

"-------------------------------------------------------------------------

state_diagram DRDY_MACHINE

" This machine generates READY for all DRAM accesses.  This output should be
" externally OR'ed with any other external ready type peripherals.

    state IDLE:
        if (RAS & !REF_CYC & MUX) then ACTIVE;
        else IDLE;

    state ACTIVE:
        if (!REF_CYC) # (RESET) then IDLE;
        else ACTIVE;
```

Table A-1. 33 MHz Simple DRAM Controller PLD Equations (Sheet 7 of 9)

```
"-------------------------------------------------------------------------------

state_diagram HOLDOFF_MACHINE

" This machine is used to edge detect the square wave refesh request.  See
" REF_REQ_MACHINE.

    state IDLE:
        if (REF_CYC) then ACTIVE;
        else IDLE;

    state ACTIVE:
        if (!R4) then IDLE;
        else ACTIVE;

"-------------------------------------------------------------------------------

state_diagram REF_REQ_MACHINE

" This machine is used to hold a pending refresh request.  R4 is a 65 KHz
" square wave derived from the refresh timer (or some other source).

    state IDLE:
        if (R4 & !HOLDOFF) then ACTIVE;
        else IDLE;

    state ACTIVE:
        if (REF_CYC) then IDLE;
        else ACTIVE;

"-------------------------------------------------------------------------------

state_diagram REF_CYC_MACHINE

" This machine indicates when the DRAM controller is actually performing
" a refresh cycle.

    state IDLE:
        if (!CPU_CYC * REF_REQ) then ACTIVE;
        else IDLE;

    state ACTIVE:
        if (!RAS & MUX) then IDLE;
        else ACTIVE;
```

Table A-1. 33 MHz Simple DRAM Controller PLD Equations (Sheet 8 of 9)

```
"-------------------------------------------------------------------------------

state_diagram RD_MACHINE

" This machine tracks all READ cycles issued by the CPU.  It assumes that
" the internal wait state generator is used, since it relies on BLAST to
" signify the end of the cycle.

    state IDLE:
        if (ADS & READ) then ACTIVE;
        else IDLE;

    state ACTIVE:
        if (BLAST) # (RESET) then IDLE;
        else ACTIVE;

"-------------------------------------------------------------------------------

state_diagram REFRESH

" This machine divides the 1.8432 MHz REFCLK by 28, producing a 65 KHz
" square wave to request refesh cycles.  If a higher frequency REFCLK is
" used, modify this machine accordingly.  If an external 65KHz source is
" available, this machine may be eliminated.  Convert node R4 into a pin
" and apply the external refresh signal to this pin

    state 0:     goto 1;
    state 1:     goto 2;
    state 2:     goto 3;
    state 3:     goto 4;
    state 4:     goto 5;
    state 5:     goto 6;
    state 6:     goto 7;
    state 7:     goto 8;
    state 8:     goto 9;
    state 9:     goto 10;
    state 10:    goto 11;
    state 11:    goto 12;
    state 12:    goto 13;
    state 13:    goto 14;
    state 14:    goto 15;
    state 15:    goto 16;
    state 16:    goto 17;
    state 17:    goto 18;
    state 18:    goto 19;
    state 19:    goto 20;
    state 20:    goto 21;
    state 21:    goto 22;
    state 22:    goto 23;
    state 23:    goto 24;
    state 24:    goto 25;
    state 25:    goto 26;
    state 26:    goto 27;
    state 27:    goto 0;
    state 28:    goto 29;
```

OUTPUT MACROCELLS		BURIED MACROCELLS	
Signal	**Product Terms**	**Signal**	**Product Terms**
$\overline{\text{RAS}}$	7	$\overline{\text{CPU\_CYC}}$	3
$\overline{\text{CAS3}}$	3	$\overline{\text{REF\_REQ}}$	2
$\overline{\text{CAS2}}$	3	$\overline{\text{REF\_CYC}}$	3
$\overline{\text{CAS1}}$	3	$\overline{\text{HOLDOFF}}$	2
$\overline{\text{CAS0}}$	3	S1	4
DRA1	3	S0	3
DRA0	3	R4	4
$\overline{\text{DRDY}}$	2	R3	5
$\overline{\text{MUX}}$	1	R2	4
$\overline{\text{WE}}$	1	R1	3
$\overline{\text{RD}}$	2	R0	2
$\overline{\text{WR}}$	1		
$\overline{\text{OE245}}$	2		

OUTPUT MACROCELLS		BURIED MACROCELLS	
Signal	Product Terms	Signal	Product Terms
$\overline{RAS}$	7	$\overline{CPU\_CYC}$	3
$\overline{CAS0}$	3	$\overline{REF\_REQ}$	2
$\overline{CAS2}$	3	$\overline{REF\_CYC}$	3
$\overline{CAS1}$	3	$\overline{HOLDOFF}$	2
$\overline{CAS0}$	3	S1	4
DRA1	3	S0	3
DRA0	3	R4	4
$\overline{DRDY}$	2	R3	5
$\overline{MUX}$	1	R2	4
$\overline{WE}$	1	R1	3
$\overline{RD}$	2	R0	2
$\overline{WR}$	1		
$\overline{OE2:5}$	2		

APPENDIX B
IMPLEMENTATION FOR SPECIFIC PLDs

Appendix A contains device-independent programmable logic equations and fitter data. Appendix B partitions the design into specific PLDs which are readily available and relatively inexpensive.

According to the fitter report (Table A-2, Signal and Product Term Allocation), the maximum number of product terms in any state machine is seven. State machines like these, which need few product terms, can fit into simple PLDs.

Most of the state logic fits into PLD1, including the state machines RAS, MUX, DRDY, CPU_CYC, REF_CYC, RD and CAS. As indicated in Table B-1, these machines share a number of input and feedback signals. In this design, address decoding is performed externally in a 74F138 which feeds the single !DRAM input. PLD1 is a 22V10 device.

PLD2 contains the burst tracking state logic, refresh request flag and combinatorial DRAM address outputs. See Table B-2. PLD2 is a 20V8 device.

PLD3, illustrated in Table B-3, contains the refresh counter and a combinational output enable for a 74F245 bus transceiver. In the example, a 1.8432 MHz clock is divided to generate the refresh timer output R4 for PLD1. PLD3 is a simple 16R6 device.

Partitioning the design into multiple simple PLDs was not tested on actual hardware.

Table B-1. PLD 1 Source Code Table (Sheet 1 of 3)

```
ABEL 5.10  -  Device Utilization Chart

960CA DRAM Controller (21121)
Rick Schue  Intel Corp.  11/21/94



-------------------------------------------------------------------------------

Module                  : 'pld1'

-------------------------------------------------------------------------------

Input files:

    ABEL PLA file       : pld1.tt3
    Device library      : P22V10.dev

Output files:

    Report file         : pld1.doc
    Programmer load file : pld1.jed
```

Table B-1. PLD 1 Source Code Table (Sheet 2 of 3)

```
P22V10 Programmed Logic:
-------------------------------------------------------------------------------

RAS.D    = (  !ADS & !DRAM & RAS.FB & MUX & REF_REQ
            #   BLAST & !RAS.FB & REF_CYC
            #   DRDY & !RAS.FB & REF_CYC
            #   RAS.FB & !CPU_CYC & REF_CYC
            #   BLAST & !RAS.FB & !CAS0
            #   DRDY & !RAS.FB & !CAS0
            #   RAS.FB & !REF_CYC & !CAS0 ); " ISTYPE 'INVERT'
RAS.C    = (  PCLK );

MUX.D    = (  RAS ); " ISTYPE 'BUFFER'
MUX.C    = (  PCLK );

CAS0.D   = (  !REF_REQ & CPU_CYC & CAS0.FB
            #   RAS & !REF_CYC & !CAS0.FB
            #   !RAS & !MUX & REF_CYC & CAS0.FB & !BE0 ); " ISTYPE 'INVERT'
CAS0.C   = (  PCLK );

CAS1.D   = (  !REF_REQ & CPU_CYC & CAS1.FB
            #   RAS & !REF_CYC & !CAS1.FB
            #   !RAS & !MUX & REF_CYC & CAS1.FB & !BE1 ); " ISTYPE 'INVERT'
CAS1.C   = (  PCLK );

CAS2.D   = (  !REF_REQ & CPU_CYC & CAS2.FB
            #   RAS & !REF_CYC & !CAS2.FB
            #   !RAS & !MUX & REF_CYC & CAS2.FB & !BE2 ); " ISTYPE 'INVERT'
CAS2.C   = (  PCLK );

CAS3.D   = (  !REF_REQ & CPU_CYC & CAS3.FB
            #   RAS & !REF_CYC & !CAS3.FB
            #   !RAS & !MUX & REF_CYC & CAS3.FB & !BE3 ); " ISTYPE 'INVERT'
CAS3.C   = (  PCLK );

DRDY.D   = (  !RAS & !MUX & REF_CYC & DRDY.FB
            #   RESET & !REF_CYC & !DRDY.FB ); " ISTYPE 'INVERT'
DRDY.C   = (  PCLK );

RD.D     = (  BLAST & RESET & !RD.FB
           #   !ADS & RD.FB & !READ ); " ISTYPE 'INVERT'
RD.C     = (  PCLK );

CPU_CYC.D    = (  CPU_CYC.FB & !ADS & !DRAM
                #   !CPU_CYC.FB & BLAST & RESET
                #   !CPU_CYC.FB & DRDY & RESET ); " ISTYPE 'INVERT'
CPU_CYC.C    = (  PCLK );

REF_CYC.D    = (  REF_REQ & REF_CYC.FB
                #   !CPU_CYC & REF_CYC.FB
                #   RAS & !MUX & !REF_CYC.FB ); " ISTYPE 'BUFFER'
REF_CYC.C    = (  PCLK );
```

Table B-1. PLD 1 Source Code Table (Sheet 3 of 3)

```
-------------------------------------------------------------------

960CA DRAM Controller (21121)
Rick Schue  Intel Corp.  11/21/94

P22V10 Chip Diagram:
-------------------------------------------------------------------

                              P22V10

                          +---------\       /---------+
                          |          \     /          |
                          |           -----           |
                   PCLK   |  1                  24    |  Vcc
                          |                           |
                    ADS   |  2                  23    |  !RAS
                          |                           |
                  BLAST   |  3                  22    |  !CAS3
                          |                           |
                  RESET   |  4                  21    |  !CAS2
                          |                           |
                   DRAM   |  5                  20    |  !CAS1
                          |                           |
                    BE0   |  6                  19    |  !CAS0
                          |                           |
                    BE1   |  7                  18    |  !DRDY
                          |                           |
                    BE2   |  8                  17    |  MUX
                          |                           |
                    BE3   |  9                  16    |  !RD
                          |                           |
                   READ   |  10                 15    |  !CPU_CYC
                          |                           |
                REF_REQ   |  11                 14    |  REF_CYC
                          |                           |
                    GND   |  12                 13    |
                          |                           |
                          `---------------------------'

                SIGNATURE: N/A
```

```
--------------------------------------------------------------------------------

Module                    : 'pld2'

--------------------------------------------------------------------------------

Input files:

    ABEL PLA file         : pld2.tt3
    Device library        : P20V8R.dev

Output files:

    Report file           : pld2.doc
    Programmer load file  : pld2.jed

P20V8R Programmed Logic:
--------------------------------------------------------------------------------


DRA1      = !(  !S1 & S0
            #    !S1 & MUX & !A12
            #    !S1 & !MUX & !A3 );

DRA0      = !(  S1 & !S0
            #    !S0 & MUX & !A11
            #    !S0 & !MUX & !A2 );

WE      = !(  READ & REF_CYC );

WR      = !(  READ & !WAIT );

REF_REQ.D    = (  REF_CYC & !REF_REQ.FB
               #    R4 & REF_REQ.FB & HOLDOFF ); " ISTYPE 'INVERT'
REF_REQ.C    = (  PCLK );

HOLDOFF.D    = (  !REF_CYC & HOLDOFF.FB
               #    !HOLDOFF.FB & R4 ); " ISTYPE 'INVERT'
HOLDOFF.C    = (  PCLK );

S1.D    = (  !A3 & !S1.FB & !S0.FB
           #    A2 & !S1.FB & !S0.FB
```

Table B-2. PLD 2 Source Code Table (Sheet 2 of 2)

```
          #    !S1.FB & DRDY
          #    !DRDY & !BLAST ); " ISTYPE 'INVERT'
S1.C    = (  PCLK );

S0.D    = !(  S0.FB & DRDY
          #    S1.FB & !DRDY & BLAST
          #    !A2 & !S0.FB & !DRDY & BLAST ); " ISTYPE 'INVERT'
S0.C    = (  PCLK );


P20V8R Chip Diagram:
---------------------------------------------------------------------------
```

P20V8R

PCLK	1	24	Vcc
BLAST	2	23	REF_CYC
RESET	3	22	!S0
A12	4	21	!S1
A11	5	20	!HOLDOFF
A3	6	19	!REF_REQ
A2	7	18	!WR
READ	8	17	!WE
WAIT	9	16	!DRA0
MUX	10	15	!DRA1
DRDY	11	14	R4
GND	12	13	

SIGNATURE: N/A

Table B-3. PLD 3 Source Code Table (Sheet 1 of 2)

```
ABEL 5.10  -  Device Utilization Chart

960CA DRAM Controller (21121)
Rick Schue  Intel Corp.  11/21/94


-------------------------------------------------------------------------------

Module                  : 'pld3'

-------------------------------------------------------------------------------

Input files:

    ABEL PLA file       : pld3.tt3
    Device library      : P16R6.dev

Output files:

    Report file         : pld3.doc
    Programmer load file : pld3.jed

P16R6 Programmed Logic:
-------------------------------------------------------------------------------

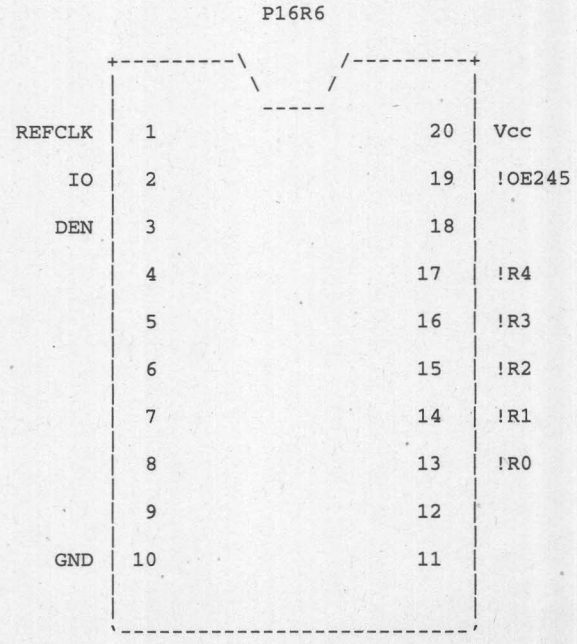

OE245     = !(  !DEN & !IO );

R4.D   = (  !R4.FB & !R3.FB
         #   R4.FB & R3.FB & R2.FB & R1.FB
         #   !R4.FB & !R1.FB
         #   R3.FB & !R2.FB & R1.FB & R0.FB
         #   !R4.FB & !R0.FB ); " ISTYPE 'INVERT'
R4.C   = (  REFCLK );

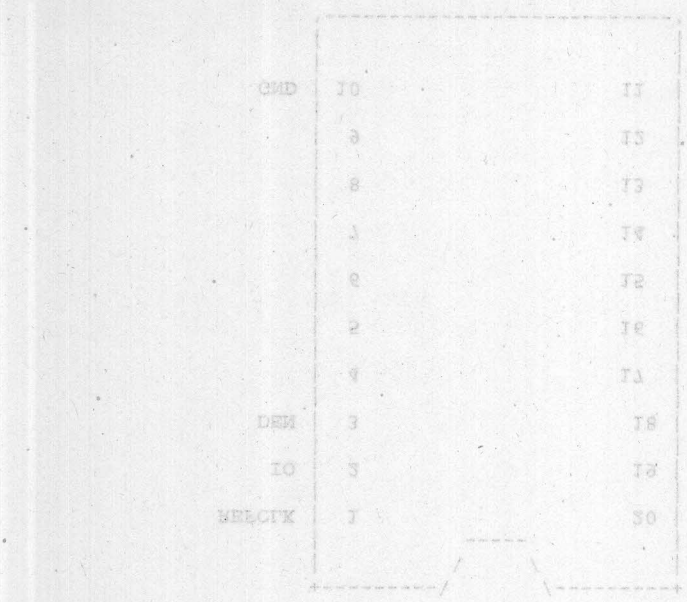
R3.D   = (  !R3.FB & !R2.FB
         #   !R3.FB & !R1.FB
         #   R4.FB & R3.FB & R1.FB & R0.FB
         #   R3.FB & R2.FB & R1.FB & R0.FB
         #   !R3.FB & !R0.FB
         #   R4.FB & R2.FB & R1.FB & !R0.FB ); " ISTYPE 'INVERT'
R3.C   = (  REFCLK );

R2.D   = (  R4.FB & R3.FB & R1.FB
         #   !R2.FB & !R1.FB
         #   R2.FB & R1.FB & R0.FB
         #   !R2.FB & !R0.FB ); " ISTYPE 'INVERT'
```

P16R6 Chip Diagram:



SIGNATURE: N/A


CMD
DEM
IO
MEMCLK

intel

AP-706

APPLICATION NOTE

DRAM Controller for 40 MHz i960® CA/CF Microprocessors

February 2, 1995

Order Number: 272655-001

AP-706

APPLICATION
NOTE

DRAM Controller for 40 MHz i960® CA/CF Microprocessors

February 2, 1995

1.0 INTRODUCTION

This application note describes a DRAM controller for use with the i960® CF 40 MHz microprocessor. Other application notes are available which describe DRAM controllers for the i960 Cx and Jx processors; see Section 7.0, RELATED INFORMATION for ordering information.

This DRAM controller's design features include:

- Interleaved design
- Can use standard 70 ns DRAM SIMM
- 4-0-1-0/3-0-1-0 back-to-back/idle bus wait state burst reads up to 40 MHz
- 3-1-1-1/2-1-1-1 back-to-back/idle bus wait state burst writes up to 40 MHz
- Single clock input to state machines

This document contains some general DRAM controller theory as well as this design's state machine definitions and timing diagrams. It also contains the PLD equations used to build and test the prototype design. All timing analysis was verified with Timing Designer*. PLD equations were generated in ABEL* as a device-independent design. Schematics were created in OrCAD*. The timing analysis, schematics and PLD files are available through Intel's America's Application Support BBS, at (916) 356-3600.

2.0 OVERVIEW

This section provides an overview of DRAM SIMM operation and the concept of memory interleaving. It also describes the i960 Cx microprocessor burst capabilities.

2.1 Page Mode DRAM SIMM Review

Page mode DRAM allows faster memory access by keeping the same row address while selecting random column addresses within that row. A new column address is selected by deasserting $\overline{CAS}$ while keeping $\overline{RAS}$ active and then asserting $\overline{CAS}$ with the new column address valid to the DRAM. Page mode operation works very well with burst buses in which a single address cycle can be followed by multiple data cycles.

All $\overline{WE}$ pins on each SIMM are tied to a common $\overline{WE}$ line; this line requires the use of early write cycles. In an early write cycle, write data is referenced to the falling edge of $\overline{CAS}$, not the falling edge of $\overline{WE}$.

Each SIMM also has four $\overline{CAS}$ lines, one for every eight (nine) bits in a 32-bit (36-bit) SIMM module. The four $\overline{CAS}$ lines control the writing to individual bytes within each SIMM.

2.2 Bank Interleaving

Interleaving significantly improves memory system performance by overlapping accesses to consecutive addresses. Two-way interleaving is accomplished by dividing the memory into two 32-bit banks (also referred to as "leaves"):

- one bank for even word addresses (A2=0)
- one bank for odd word addresses (A2=1)

The two banks are read in parallel and the data from the two banks is multiplexed onto the processor's data bus. This overlaps the wait states of:

- the second access with the first
- the third access with the second
- the fourth access with the third

Figure 1 shows DRAM with a 2-1-1-1 quad word burst read wait state profile being interleaved to generate a 2-0-0-0 wait state system.

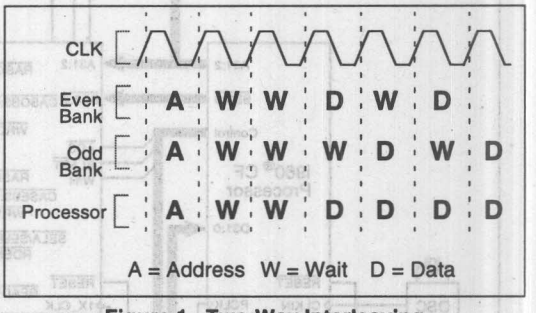


Figure 1. Two-Way Interleaving

2.3 Burst Capabilities for 32-Bit Bus

A bus access starts by asserting $\overline{ADS}$ in the address cycle, and ends by asserting $\overline{BLAST}$ in the last data cycle. Figure 2 shows $\overline{ADS}$ and $\overline{BLAST}$ timings for a quad-word access. i960 Cx processor's burst protocol requires:

- Quad-word and triple-word requests always start on quad word boundaries (A3 = 0, A2 = 0).
- Double-word requests always start on double word boundaries (A3 = X, A2 = 0).

SIMMS. This section describes each block.

3.1 Control Logic

The DRAM controller is centered around a four-bit state machine which controls DRAM bank accesses and refreshing. All timings are generated based on the four-bit state machine's outputs. Some states are used for both read and write accesses. The state machine uses the W_$\overline{R}$ signal from the processor to distinguish between reads and writes.

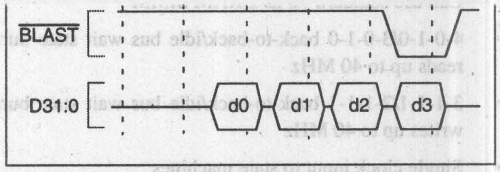


Figure 2. Quad-Word Access Example Showing $\overline{ADS}$ and $\overline{BLAST}$ Timings

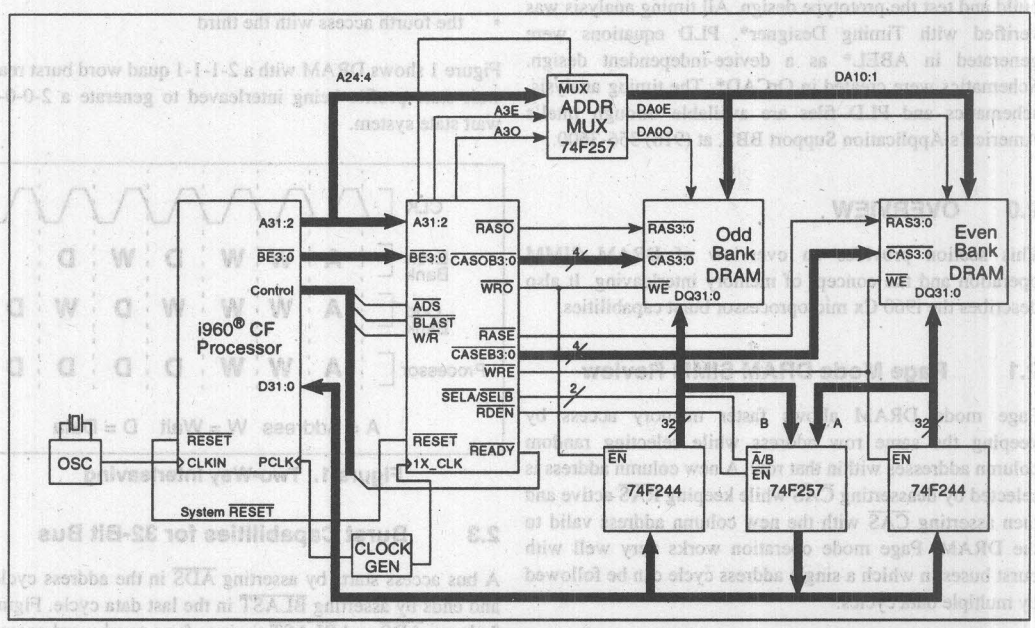


Figure 3. DRAM Controller Block Diagram

3.1.1 Refresh Logic

Typically DRAMs need to be refreshed every 15.6 µs. In this design, due to power requirements needed to refresh an entire DRAM array, one bank is refreshed at a time. The DRAM controller uses an eight-bit counter to generate refresh requests. A refresh request is generated every 7.8 µs. The DRAM controller toggles between refreshing each bank every 7.8 µs which means each bank is effectively refreshed every 15.6 µs.

A refresh request has priority over a processor request. When a processor and a refresh request occur simultaneously, the DRAM controller sequences a refresh to the appropriate DRAM bank while the PENDING state machine posts the processor request. The pending request is then serviced after the refresh is completed.

An eight-bit synchronous down counter is used to generate refresh requests. The counter is clocked using the 1X_CLK clock. The $\overline{\text{REFREQ}}$ signal is asserted each time the counter reaches zero. Counting is inhibited when the counter reaches zero. The counter is reloaded with 0xff and counting resumes after the ACCESS state machine services the refresh. During reset, the counter is loaded with 0xff.

3.1.2 Clock Generation

A Motorola* MC88915 low skew CMOS PLL generates the clock signals for the DRAM controller. The MC88915 uses PCLK2 as an input, and produces four very low skew copies of PCLK2, as well as a 2x PCLK. At 40 MHz, the maximum skew between PCLK2 and any of the MC88915 outputs was calculated to be ±1 ns, while the skew between any of the individual outputs is ±750 ps under equal loading conditions. All clock lines are terminated with 22 ohm series resistors.

3.1.3 Wait State Profile

The DRAM Controller uses the processor's $\overline{\text{READY}}$ signal to control wait states. The MCON register is initialized as follows: ($N_{XAD} = N_{XDA} = N_{XDD} = N_{XDA} = 0$). Table 1, Wait State Profiles, provides the wait state profiles for read and write accesses up to 40 MHz.

Back-to-back accesses require an extra wait state to meet RAS precharge time. Therefore, to meet the RAS precharge time required, the first data access for reads uses four wait state cycles as opposed to three wait state cycles for idle bus DRAM accesses. For writes, the first data access uses three wait state cycles for back-to-back accesses, as opposed to two wait state cycles for idle bus.

Table 1. Wait State Profiles (40 MHz)

Access Type	Wait State Profile	
	Back-To-Back	Idle Bus
Quad Word Read	4-0-1-0	3-0-1-0
Triple Word Read	4-0-1	3-0-1
Double Word Read	4-0	3-0
Single Word Read	4	3
Quad Word Write	3-1-1-1	2-1-1-1
Triple Word Write	3-1-1	2-1-1
Double Word Write	3-1	2-1
Single Word Write	3	2

3.2 Address Path

Figure 4 illustrates a block diagram of the address path logic. The 2-to-1 multiplexers combine the row and column addresses into a singular row/column address that the DRAM requires. DA0E and DA0O equivalent signals are generated, one for each bank. DA0E and DA0O are generated by using A3E and A3O respectively. DA0E and DA0O are the only address bits that increment during bursts. The timing of these signals during bursts is critical for proper operation.

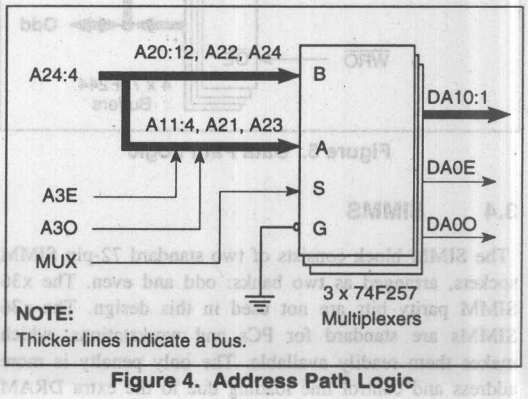


Figure 4. Address Path Logic

3.3 Data Path

As shown in Figure 5, there is one data path for reads and a separate data path for writes. The read path uses 74F257 2:1 multiplexers to prevent contention between the two DRAM banks. $\overline{CAS}$ can be active for both banks at the same time, necessitating use of the multiplexers. The multiplexer outputs are enabled only during reads by the $\overline{RDEN}$ signal. The multiplexers are switched using $\overline{SELA}$ and $\overline{SELB}$. These signals are derived based on the states of the ACCESS state machine and address A2. The write data path consists of eight 8-bit 74F244 buffers, four for each bank. The buffer outputs are enabled by $\overline{WRE}$ and $\overline{WRO}$.

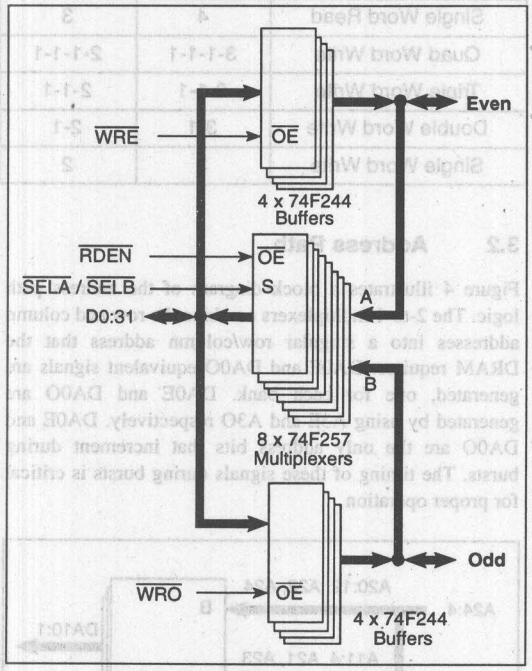


Figure 5. Data Path Logic

3.4 SIMMS

The SIMM block consists of two standard 72-pin SIMM sockets, arranged as two banks: odd and even. The x36 SIMM parity bits are not used in this design. The x36 SIMMs are standard for PCs and workstations, which makes them readily available. The only penalty is more address and control line loading due to the extra DRAM devices of the x36 SIMM. All address and control lines to the SIMMs are terminated with 22 ohm resistors.

4.0 STATE MACHINES AND SIGNALS

This section describes the state machines and signals used in this design. Most of the state machines are simple and the PLD equations can be referenced in APPENDIX A. The ACCESS state machine is the most complex of all the state machines; for that reason, this application note provides more detail on the operations of this state machine. In this design, all the state machines are clocked using 1X_CLK clock (bus clock).

All PLD equations are written in ABEL. APPENDIX A, PLD EQUATIONS contains a listing of the PLD equation file. The state machine transitions described here follow the ABEL conventions for logic operators.

- ! represents NOT, bit-wise negation
- & represents AND
- # represents OR

To follow the ABEL conventions, active LOW signals (such as ADS) already have a polarity assigned. For example, in the state machines, ADS refers to the asserted state (LOW) and !ADS refers to the non-asserted state (HIGH).

4.1 ACCESS State Machine

The ACCESS state machine, the "heart" of the DRAM controller, is implemented as a four-bit state machine. See Figure 6, Basic ACCESS State Machine. It is responsible for sequencing accesses as well as refreshes to the DRAM banks.

From the IDLE state, the access state machine is sequenced based on these three events:

- Refresh requests from the counter
- DRAM requests from the processor
- PENDING state machine requests

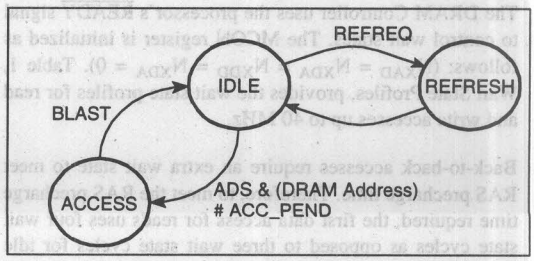


Figure 6. Basic ACCESS State Machine

4.2 PENDING State Machine

The PENDING state machine is a one-bit state machine which monitors DRAM requests from the processor. This is necessary because a DRAM refresh has priority over a processor request. Therefore, this state machine is used to post the processor request. The state machine gets reset once the ACCESS state machine starts sequencing the pending request. The state machine generates $\overline{\text{ACC\_PEND}}$.

4.3 ODDACCESS State Machine

The ODDACCESS state machine is clocked using the 1X_CLK clock. It is a one-bit state machine which monitors the initial state of the processor's address A2. Several state machines in this design use the output of this state machine as inputs. Address A2 from the processor indicates whether an access starts on an even or odd word boundary. The ACCESS state machine uses this bit extensively. It is important to latch address A2 because the processor toggles address A2 on burst accesses. This state machine generates $\overline{\text{LA2}}$.

4.4 BANKSELA State Machine

The BANKSELA state machine is a one-bit state machine which is used to control the data multiplexer, primarily to select between even or odd data during read accesses. This state machine is clocked using the 1X_CLK clock. It generates $\overline{\text{SELA}}$.

4.5 BANKSELB State Machine

The BANKSELB state machine is a one-bit state machine which is used to control the data multiplexer, primarily to select between even or odd data during read accesses. This state machine is clocked using the 1X_CLK clock. It generates $\overline{\text{SELB}}$.

4.6 ADDRMUX State Machine

The ADDRMUX state machine is a one-bit state machine which is used to control the address multiplexers, primarily to select between row or column addresses. It is clocked using the 1X_CLK clock. This state machine generates $\overline{\text{MUX}}$, which is a delayed version of $\overline{\text{RASE}}$. By delaying the switching of the row address by one 1X_CLK clock cycle provides ample row address hold time (t_{RAH}) required by the DRAM. The row address is selected while $\overline{\text{MUX}}$ is high; otherwise, the column address is selected.

4.7 A3EVEN State Machine

The A3EVEN state machine is a one-bit state machine which is toggled on burst accesses to select the next data word (next column data). The state machine is initially loaded with the value of the processor's address A3 and then toggled for the next data access. This state machine is clocked using 1X_CLK clock, and generates A3E. This signal is an input to the address multiplexer.

4.8 A3ODD State Machine

The A3ODD state machine is a one-bit state machine and has the same functionality as the A3EVEN state machine. This state machine generates A3O.

4.9 RFEVENBK State Machine

The RFEVENBK state machine is a one-bit state machine which is used to indicate which of the two banks (even or odd) to refresh. The two banks are refreshed separately. The even bank is refreshed when the RFEVENBK state machine is active; otherwise, the odd bank is refreshed. The output of this state machine is toggled on every refresh. This state machine generates $\overline{\text{REFEVEN}}$.

4.10 CASPIPE State Machine

The CASPIPE state machine is a one-bit state machine which generates a pipelined $\overline{\text{CAS}}$ signal one 1X_CLK clock cycle earlier. The output of this state machine is then fed to the CASE_B3:0 state machines where it is reconstructed to drive the $\overline{\text{CAS}}$ lines of the even bank. This state machine generates $\overline{\text{CASEE}}$.

4.11 CASPIPO State Machine

The CASPIPO state machine is a one-bit state machine which generates a pipelined $\overline{\text{CAS}}$ signal one 1X_CLK clock cycle earlier. The output of this state machine is then fed to the CASO_B3:0 state machines where it is reconstructed to drive the $\overline{\text{CAS}}$ lines of the odd bank. This state machine generates $\overline{\text{CASOO}}$.

4.12 CASE_B3:0 State Machines

The CASE_B3:0 state machines control the $\overline{\text{CAS}}$ pins of the even bank. CASEB0 controls the least significant byte and $\overline{\text{CASEB3}}$ controls the most significant byte. The CASE_B0 state machine generates $\overline{\text{CASEB0}}$, and the CASE_B3 state machine generates $\overline{\text{CASEB3}}$. $\overline{\text{CASEB0}}$ is

CASO_B3 state machine generates CASOB3. CASOB0 is asserted when CASOO and the processor's BE0 signal are asserted. CASOB3 is asserted when CASOO and the processor's BE3 signal are asserted. The CASO_B3:0 state machines are clocked using the 1X_CLK clock.

4.14 REFREQ Signal

REFREQ, an active low signal, is the output of an eight-bit counter. The counter is clocked using the 1X_CLK clock. REFREQ is asserted when the counter reaches zero. The ACCESS state machine uses REFREQ to sequence refreshes.

4.15 RASEVEN State Machine

The RASEVEN state machine is a one-bit state machine which is used to generate RAS signals for the even bank. It is clocked using the 1X_CLK clock. This state machine generates RASE.

4.16 RASODD State Machine

The RASODD state machine is a one-bit state machine which is used to generate RAS signals for the odd bank. It is clocked using the 1X_CLK clock. This state machine generates RASO.

4.17 SRASE State Machine

The SRASE state machine is a one-bit state machine which is used to monitor back-to-back DRAM accesses. It is generated by shifting RASE by one 1X_CLK clock cycle. This state machine generates SRASE. By using the state of this signal the DRAM controller can eliminate one wait state cycle for accessing the first data word. Back-to-back accesses require an extra wait state cycle to satisfy the RAS precharge time (t_{RP}).

4.20 WRO Signal

WRO is asserted while a DRAM write is in progress. It controls the odd leaf WE lines to perform early writes. It also controls the odd data path buffers output enables.

5.0 DRAM CONTROLLER ACCESS FLOW

This section explains how the ACCESS state machine is sequenced while reading, writing, and refreshing DRAMs. Examples used are:

- quad-word read
- single-word read
- quad-word write
- single-word write
- refresh

The examples in this document assume back-to-back DRAM accesses or pending accesses. For example, the first data access of a DRAM request uses four wait states for reads and three wait states for writes. For idle bus accesses, the ACCESS0 state is skipped, allowing only three wait states for reads and two wait states for writes. Refer to the PLD equations in APPENDIX A. The ACCESS state machine uses SRASE to detect back-to-back accesses.

RDEN is asserted during read accesses while WRE and WRO are asserted during write accesses.

5.1 Quad-Word Read

Figure 7 shows the state diagram for a quad-word read; Figure 8 shows the timing diagram. This state diagram also shows the state machine paths for triple-, double-, and single-word reads. Single-word reads which are aligned on odd word boundaries use a different path; therefore, a separate example is used to explain that state machine path.

From the IDLE state, the machine enters the ACCESS0 state due to a processor request or a pending processor request. At the end of the IDLE state, the A3EVEN and A3ODD state machines are loaded with the processor's address A3 and the ODDACCESS state machine is loaded with the processor's address A2. While in the IDLE state, $\overline{MUX}$ is deasserted, which selects the row address.

At the end of the ACCESS0 state, the $\overline{RASE}$ and $\overline{RASO}$ are asserted. The machine then proceeds to the ACCESS1 state.

At the end of the ACCESS1 state, $\overline{MUX}$ is asserted. This causes the column address to be selected. $\overline{CASEE}$ is asserted, and the PENDING state machine is reset. From ACCESS1, the machine enters ACCESS2 state.

At the end of the ACCESS2 state, $\overline{CASEB}$3:0 are asserted if $\overline{CASEE}$ and the respective Byte Enable signals from the processor are asserted. $\overline{CASOO}$ is asserted if $\overline{BLAST}$ is not asserted. The machine then proceeds to the ACCESS3 state.

At the end of the ACCESS3 state, $\overline{CASOB}$3:0 are asserted if $\overline{CASOO}$ and the respective Byte Enable signals from the processor are asserted, while $\overline{CASEE}$ is deasserted. The machine then proceeds to the ACCESS4 state.

The ACCESS4 state is the first or third data cycle (T_{d0}/T_{d2}) for read accesses aligned on even word boundaries. At the end of ACCESS4, $\overline{CASEE}$ is reasserted while $\overline{CASEB}$3:0 are deasserted. $\overline{CASEB}$3:0 are deasserted because $\overline{CASEE}$ is sampled deasserted. $\overline{CASOO}$ is deasserted before leaving this state. From here, the machine can proceed to either the ACCESS5 or the IDLE state. If $\overline{BLAST}$ is asserted, the machine proceeds to the IDLE state; otherwise, to the ACCESS5 state.

The ACCESS5 state is the second or fourth data cycle (T_{d1}/T_{d3}) for read accesses. At the end of the ACCESS5 state, $\overline{CASOO}$ is reasserted. $\overline{CASEB}$3:0 are asserted if $\overline{CASEE}$ and the respective Byte Enable signals from the processor are asserted. $\overline{CASOB}$3:0 are deasserted because $\overline{CASOO}$ is sampled deasserted. From here, the machine can proceed to either the ACCESS3 or the IDLE state. If $\overline{BLAST}$ is asserted, the machine proceeds to the IDLE state; otherwise, to the ACCESS3 state. The machine then proceeds to ACCESS3 state.

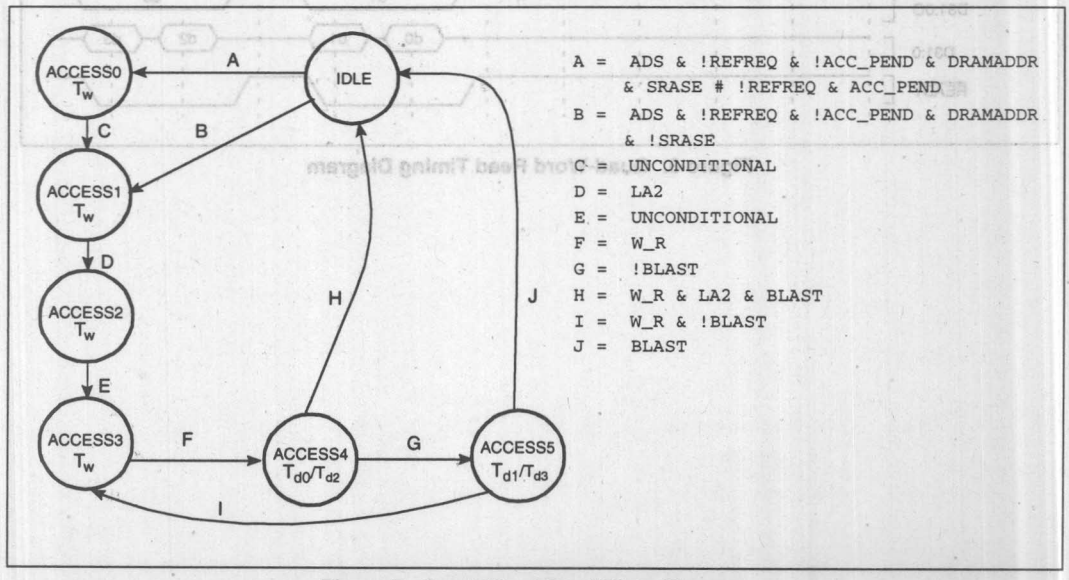


Figure 7. Quad-Word Read State Diagram

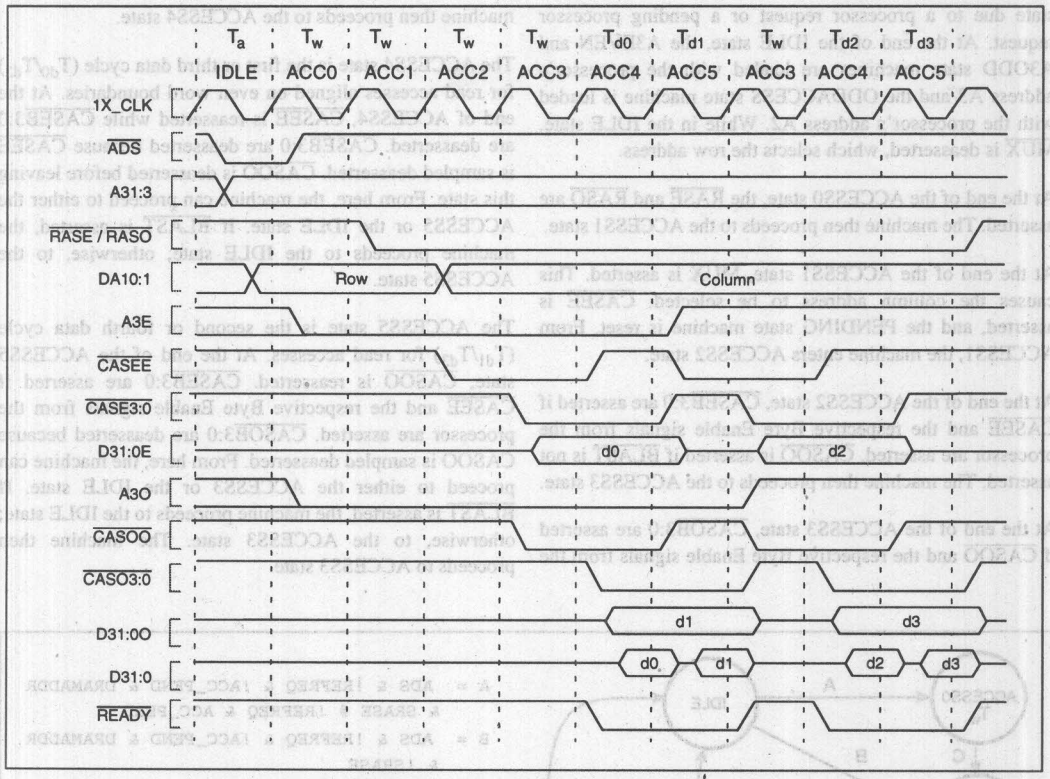


Figure 8. Quad-Word Read Timing Diagram

5.2 Single-Word Read

The ACCESS state machine takes a slightly different path when a read request is aligned on an odd word boundary. Figure 9 shows the state diagram for a single-word read; Figure 10 shows the timing diagram.

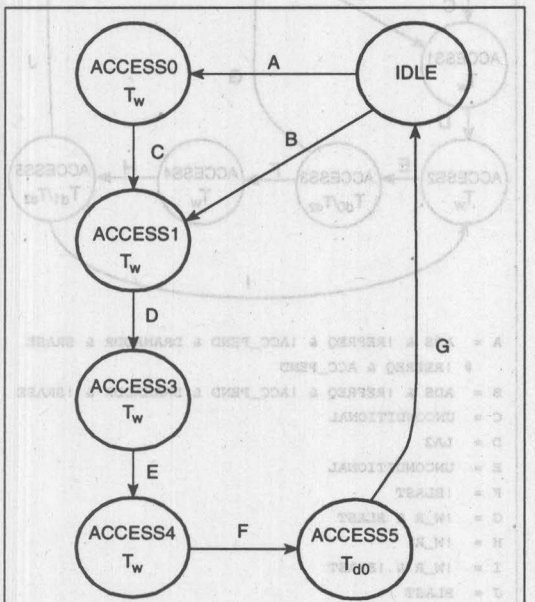


Figure 9. Single-Word Read State Diagram (A2 = 1)

From the IDLE state, the machine enters the ACCESS0 state due to a processor request or a pending processor request. At the end of the IDLE state, the A3EVEN and A3ODD state machines are loaded with the processor's address A3 and the ODDACCESS state machine is loaded with the processor's address A2. $\overline{MUX}$ is deasserted in the IDLE state, which selects the row address.

At the end of the ACCESS0 state, $\overline{RASO}$ is asserted. The machine then proceeds to the ACCESS1 state.

At the end of ACCESS1, $\overline{MUX}$ is asserted. This causes the column address to be selected. $\overline{CASOO}$ is asserted. The PENDING state machine is reset before the machine proceeds to the ACCESS3 state.

At the end of the ACCESS3 state, $\overline{CASOB}$3:0 are asserted if $\overline{CASOO}$ and the respective Byte Enable signals from the processor are asserted. The machine then proceeds to ACCESS4 state.

At the end of the ACCESS4 state, $\overline{CASOO}$ is deasserted and the machine proceeds to the ACCESS5 state.

The ACCESS5 state is the data cycle for the read access. $\overline{CASOB}$3:0 are deasserted. This is primarily because $\overline{CASOO}$ is sampled deasserted. The machine then proceeds to the IDLE state while deasserting $\overline{RASO}$.

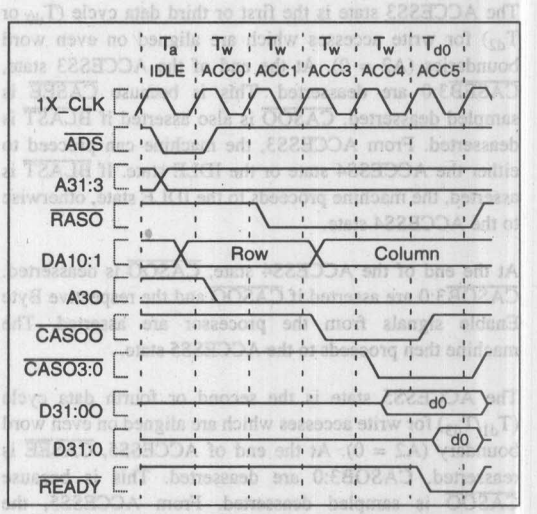


Figure 10. Single-Word Read Timing Diagram (A2 = 1)

5.3 Quad-Word Write

Figure 11 shows the state diagram for a quad-word write. This state diagram also shows the state machine paths for triple-, double-, and single-word writes. Single-word writes which are aligned on odd word boundaries use a different path, therefore, a different example is used to explain the state machine path.

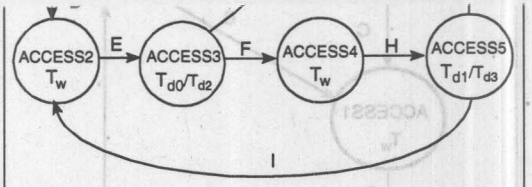
From the IDLE state, the machine enters the ACCESS0 state due to a processor request or a pending processor request. At the end of the IDLE state, the A3EVEN and A3ODD state machines are loaded with the processor's

At the end of the ACCESS2 state $\overline{\text{CASEE}}$ is deasserted. $\overline{\text{CASEB}}$3:0 are asserted if $\overline{\text{CASEE}}$ and the respective Byte Enable signals from the processor are asserted. The machine then proceeds to the ACCESS3 state.

The ACCESS3 state is the first or third data cycle (T_{d0} or T_{d2}) for write accesses which are aligned on even word boundaries (A2 = 0). At the end of the ACCESS3 state, $\overline{\text{CASEB}}$3:0 are deasserted. This is because $\overline{\text{CASEE}}$ is sampled deasserted. $\overline{\text{CASOO}}$ is also asserted if $\overline{\text{BLAST}}$ is deasserted. From ACCESS3, the machine can proceed to either the ACCESS4 state or the IDLE state. If $\overline{\text{BLAST}}$ is asserted, the machine proceeds to the IDLE state, otherwise to the ACCESS4 state.

At the end of the ACCESS4 state, $\overline{\text{CASOO}}$ is deasserted. $\overline{\text{CASOB}}$3:0 are asserted if $\overline{\text{CASOO}}$ and the respective Byte Enable signals from the processor are asserted. The machine then proceeds to the ACCESS5 state.

The ACCESS5 state is the second or fourth data cycle (T_{d1}/T_{d3}) for write accesses which are aligned on even word boundary (A2 = 0). At the end of ACCESS5, $\overline{\text{CASEE}}$ is reasserted. $\overline{\text{CASOB}}$3:0 are deasserted. This is because $\overline{\text{CASOO}}$ is sampled deasserted. From ACCESS5, the machine can proceed to either the ACCESS2 state or the IDLE state. If $\overline{\text{BLAST}}$ is asserted, the machine proceeds to the IDLE state, otherwise to the ACCESS2 state.




```
A =  ADS & !REFREQ & !ACC_PEND & DRAMADDR & SRASE
   # !REFREQ & ACC_PEND
B =  ADS & !REFREQ & !ACC_PEND & DRAMADDR & !SRASE
C =  UNCONDITIONAL
D =  LA2
E =  UNCONDITIONAL
F =  !BLAST
G =  !W_R & BLAST
H =  !W_R
I =  !W_R & !BLAST
J =  BLAST
```

Figure 11. Quad-Word Write State Diagram

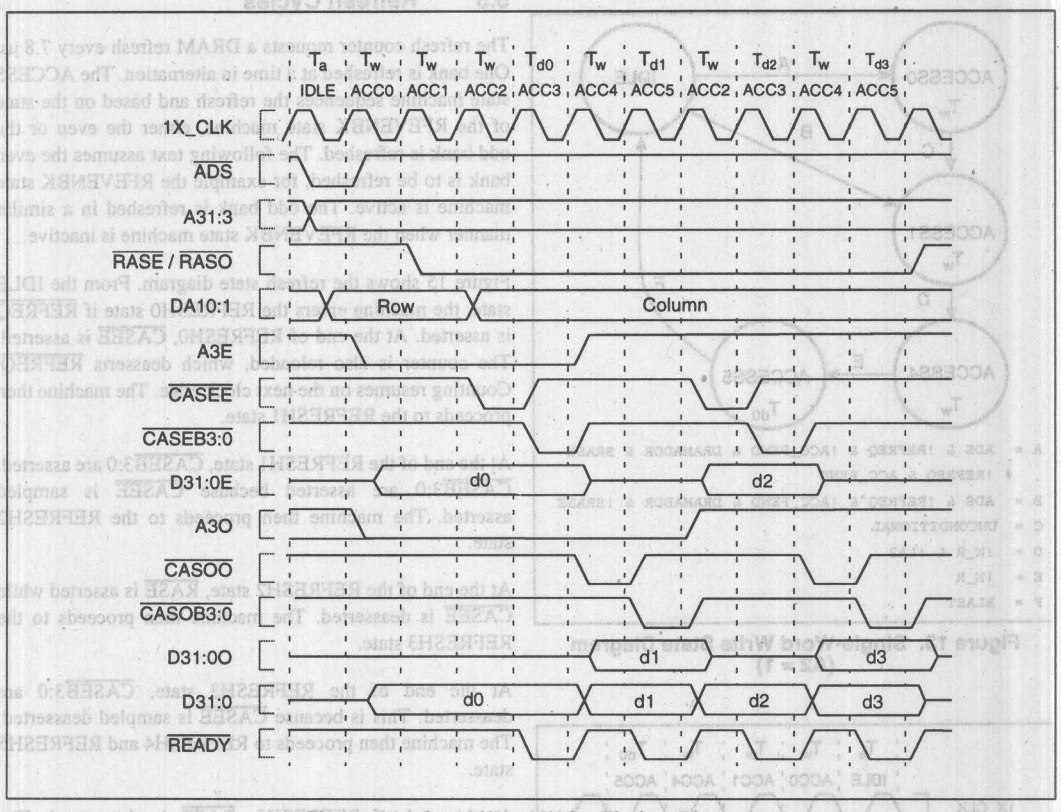


Figure 12. Quad-Word Write Timing Diagram

5.4 Single-Word Write

The ACCESS state machine takes a slightly different path when the write request is aligned on an odd word boundary. Figure 13 shows the state diagram for a single-word write; Figure 14 shows the timing diagram.

From the IDLE state, the machine enters the ACCESS0 state due to a processor request or a pending processor request. At the end of the IDLE state, the A3EVEN and A3ODD state machines are loaded with the processor's address A3, and the ODDACCESS state machine is loaded with the processor's address A2.

At the end of ACCESS0 state, $\overline{\text{RASO}}$ is asserted. The machine then proceeds to the ACCESS1 state.

At the end of ACCESS1, $\overline{\text{MUX}}$ is asserted. This causes the column address to be selected. $\overline{\text{CASOO}}$ is also asserted. From ACCESS1, the machine enters ACCESS4 state.

At the end of the ACCESS4 state, $\overline{\text{CASOB}}$3:0 are asserted if $\overline{\text{CASOO}}$ and the respective Byte Enable signals from the processor are asserted. $\overline{\text{CASOO}}$ is deasserted at the end of ACCESS4. The machine then proceeds to ACCESS5 state.

The ACCESS5 state is the data cycle (T_{d0}) for the write access which is aligned on odd word boundary (A2 = 1). At the end of ACCESS5, $\overline{\text{CASOB}}$3:0 are deasserted. The machine then proceeds to the IDLE state.

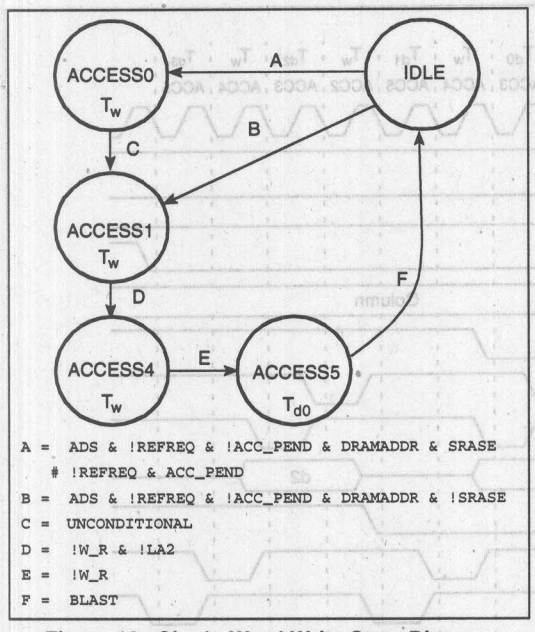


Figure 13. Single-Word Write State Diagram (A2 = 1)

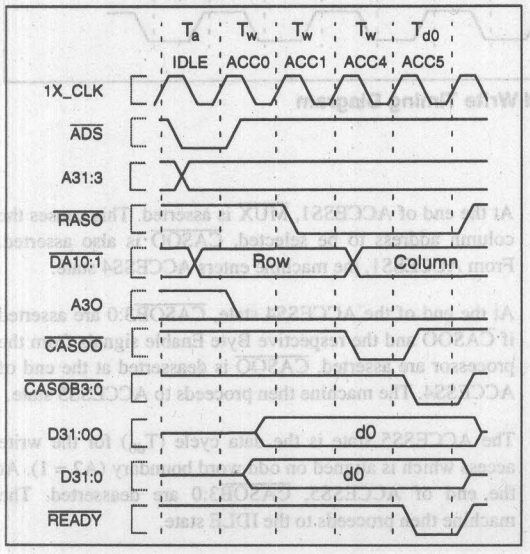


Figure 14. Single-Word Write Timing Diagram (A2 = 1)

5.5 Refresh Cycles

The refresh counter requests a DRAM refresh every 7.8 µs. One bank is refreshed at a time in alternation. The ACCESS state machine sequences the refresh and based on the state of the RFEVENBK state machine, either the even or the odd bank is refreshed. The following text assumes the even bank is to be refreshed, for example the RFEVENBK state machine is active. The odd bank is refreshed in a similar manner when the RFEVENBK state machine is inactive.

Figure 15 shows the refresh state diagram. From the IDLE state, the machine enters the REFRESH0 state if $\overline{\text{REFREQ}}$ is asserted. At the end of REFRESH0, $\overline{\text{CASEE}}$ is asserted. The counter is also reloaded, which deasserts $\overline{\text{REFREQ}}$. Counting resumes on the next clock edge. The machine then proceeds to the REFRESH1 state.

At the end of the REFRESH1 state, $\overline{\text{CASEB}}$3:0 are asserted. $\overline{\text{CASEB}}$3:0 are asserted because $\overline{\text{CASEE}}$ is sampled asserted. The machine then proceeds to the REFRESH2 state.

At the end of the REFRESH2 state, $\overline{\text{RASE}}$ is asserted while $\overline{\text{CASEE}}$ is deasserted. The machine then proceeds to the REFRESH3 state.

At the end of the REFRESH3 state, $\overline{\text{CASEB}}$3:0 are deasserted. This is because $\overline{\text{CASEE}}$ is sampled deasserted. The machine then proceeds to REFRESH4 and REFRESH5 state.

At the end of REFRESH5, $\overline{\text{RASE}}$ is deasserted. The machine then proceeds to the IDLE state

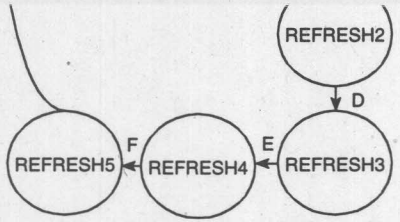


Figure 15. Refresh State Diagram

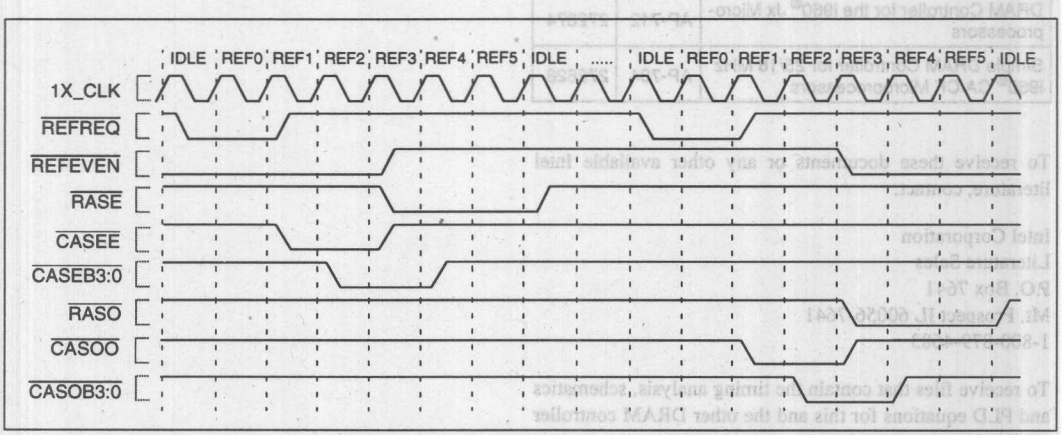


Figure 16. Refresh Timing Diagram

6.0 CONCLUSION

In conclusion, this application note describes a DRAM controller for use with i960® CF 40 MHz microprocessors. This DRAM Controller was built and tested for validation purposes. The PLD equations which were used to build and test the prototype design were created in ABEL. All timing analysis was verified with Timing Designer. Schematics were created with OrCAD. The timing analysis, schematics and PLD files are available through Intel's America's Application Support BBS, at (916) 356-3600.

7.0 RELATED INFORMATION

This application note is one of four that are related to DRAM controllers for the i960 processors. The following table shows the documents and order numbers:

Document Name	App. Note #	Order #
DRAM Controller for the 33, 25, and 16 MHz i960® CA/CF Microprocessors	AP-703	272627
DRAM Controller for the i960® Jx Microprocessors	AP-712	272674
Simple DRAM Controller for 25/16 MHz i960® CA/CF Microprocessors	AP-704	272628

To receive these documents or any other available Intel literature, contact:

Intel Corporation
Literature Sales
P.O. Box 7641
Mt. Prospect IL 60056-7641
1-800-879-4683

To receive files that contain the timing analysis, schematics and PLD equations for this and the other DRAM controller application notes, contact:

Intel Corporation
America's Application Support BBS
916-356-3600

APPENDIX A
PLD EQUATIONS

Table A-1 contains the PLD equations which were used to build and test the prototype design. Table A-2 defines signal and product term allocation. The PLD equations were created in ABEL as a device-independent design. Using the ABEL software*, a PDS file was created and subsequently imported into PLDSHELL software*. PLDSHELL was used to incorporate the design into the Altera EPX780 FLEXlogic PLD*. PLDSHELL was also used to create the JEDEC file, and to simulate the design. In addition, this appendix contains a table listing the number of product terms used by each macrocell.

This DRAM controller does not use the $\overline{\text{APK\_ACTIVE}}$ signal.

Table A-1. 40 MHz DRAM Controller PLD Equation (Sheet 1 of 19)

```
Module          CX40T
Title           'DRAM Controller for 40MHz
Source File     CX40T.ABL
Revision        Rev 0.0
Date            11/17/94
Designer        Sailesh Bissessur
                Intel i960 Applications Engineering'
"-------------------------------------------------------------------------
" 2-Way Interleaved DRAM controller for the 960CF 40MHz.
" This design also contains logic for FLASH, HEX DISPLAY, and Software Reset
"       DRAM            -  0xA0000000
"       FLASH           -  0xFFFE0000
"       HEX DISPLAY     -  0xB8000000
"       SW Reset        -  0xB0000000
"-------------------------------------------------------------------------
"        Uxx     device  'iFX780_132';
" inputs
        CLK1            PIN; " 1x clock
        CLK2            PIN;

        A2              PIN; " Address A2
        !EXTRST         PIN; " External Reset
        !CPUWAIT        PIN; " Processor Wait
        !ADS            PIN; " Address Strobe
        !BLAST          PIN; " Burst Last
        !W_R            PIN; " Read/Write
        A31             PIN; " Address A31
        A30             PIN; " Address A30
        A29             PIN; " Address A29
        A28             PIN; " Address A28
        A27             PIN; " Address A27
        DCLK1           PIN; " Delayed Clock
        A3              PIN; " Address A3
        !BE3            PIN; " Byte Enable 3
        !BE2            PIN; " Byte Enable 2
        !BE1            PIN; " Byte Enable 1
        !BE0            PIN; " Byte Enable 0
        !APK_ACTIVE     PIN; " Indicate presence of ApLink
```

```
        !REFEVEN        PIN  istype 'reg'; " which bank to refresh
        !CASEE          PIN  istype 'reg'; " Pipelined Even CAS
        !CASOO          PIN  istype 'reg'; " Pipelined Odd CAS
        A3O             PIN  istype 'reg'; " Odd Address Counter
        !WAIT           PIN  istype 'com'; " wait state indicator
        !RASE           PIN  istype 'reg'; " Even RAS
        !MUX            PIN  istype 'reg'; " Selects Row/Column Address
        !CASEB0         PIN  istype 'reg'; " Byte 0 Even CAS
        !CASEB1         PIN  istype 'reg'; " Byte 1 Even CAS
        !CASEB2         PIN  istype 'reg'; " Byte 2 Even CAS
        !CASEB3         PIN  istype 'reg'; " Byte 3 Even CAS
        !CASOB0         PIN  istype 'reg'; " Byte 0 Odd CAS
        !CASOB1         PIN  istype 'reg'; " Byte 1 Odd CAS
        !CASOB2         PIN  istype 'reg'; " Byte 2 Odd CAS
        !CASOB3         PIN  istype 'reg'; " Byte 3 Odd CAS
        S3              PIN  istype 'reg'; " Refresh Counter 1 bit 3
        S2              PIN  istype 'reg'; " Refresh Counter 1 bit 2
        S1              PIN  istype 'reg'; " Refresh Counter 1 bit 1
        S0              PIN  istype 'reg'; " Refresh Counter 1 bit 0
        T3              PIN  istype 'reg'; " Refresh Counter 2 bit 3
        T2              PIN  istype 'reg'; " Refresh Counter 2 bit 2
        T1              PIN  istype 'reg'; " Refresh Counter 2 bit 1
        T0              PIN  istype 'reg'; " refresh Counter 2 bit 0
        !REFREQ         PIN  istype 'com'; " Refresh Required
        !FLASHCS        PIN  istype 'reg'; " FLASH Chip Select
        !FLASHRD        PIN  istype 'reg'; " FLASH OE
        !FLASHWR        PIN  istype 'com'; " FLASH WE
        !XCROE          PIN  istype 'reg'; " XCR OE control
        !XCRDIR         PIN  istype 'com'; " XCR DIR control
        !SWRST          PIN  istype 'reg'; " SW Reset Indicator
        !TRIGRST        PIN  istype 'reg'; " Triggers the 7705
        !RESET          PIN  istype 'reg'; " System Reset
        !WRE            PIN  istype 'com'; " Odd Bank WE
        !WRO            PIN  istype 'com'; " Even Bank WE
        !SRASE          PIN  istype 'reg'; " Shifted RASE
        LED_LAT         PIN  istype 'reg'; " Hex Display Pulse
"
C = .C.;
X = .X.;

"
```

Table A-1. 40 MHz DRAM Controller PLD Equation (Sheet 3 of 19)

```
CYCLE       = [Q3,Q2,Q1,Q0];
ODDACCESS   = [LA2];
BANKSELA    = [SELA];
BANKSELB    = [SELB];
PENDING     = [ACC_PEND];
RDY         = [READY];
RASEVEN     = [RASE];
RASODD      = [RASO];
CASPIPE     = [CASEE];
CASPIPO     = [CASOO];
ADDRMUX     = [MUX];
A3EVEN      = [A3E];
A3ODD       = [A3O];
RFEVENBK    = [REFEVEN];
CASE_B0     = [CASEB0];
CASE_B1     = [CASEB1];
CASE_B2     = [CASEB2];
CASE_B3     = [CASEB3];
CASO_B0     = [CASOB0];
CASO_B1     = [CASOB1];
CASO_B2     = [CASOB2];
CASO_B3     = [CASOB3];
REFCT2 = [T3,T2,T1,T0];
REFCT1 = [S3,S2,S1,S0];
DRAMADDR  = (A31 & !A30 &  A29 & !A28 & !A27);
FLASHADDR = (A31 &  A30 &  A29 &  A28 &  A27);
SWRSTADDR = (A31 & !A30 &  A29 &  A28 & !A27);
LEDADDR   = (A31 & !A30 &  A29 &  A28 &  A27);

"
ASSERT   = ^b1;
DEASSERT = ^b0;

Z0       = ^b0000;
Z1       = ^b0001;
Z2       = ^b0010;
Z3       = ^b0011;
Z4       = ^b0100;
Z5       = ^b0101;
Z6       = ^b0110;
Z7       = ^b0111;
Z8       = ^b1000;
Z9       = ^b1001;
Z10      = ^b1010;
Z11      = ^b1011;
Z12      = ^b1100;
Z13      = ^b1101;
Z14      = ^b1110;
Z15      = ^b1111;
```

Table A-1. 40 MHz DRAM Controller PLD Equation (Sheet 4 of 19)

```
IDLE          = ^b0000;
ACCESS0       = ^b0001;
ACCESS1       = ^b0010;
ACCESS2       = ^b0011;
ACCESS3       = ^b0100;
ACCESS4       = ^b0101;
ACCESS5       = ^b0110;
ACCESS6       = ^b0111; "this state is never entered
ACCESS7       = ^b1000; "this state is never entered
REFRESH0      = ^b1001;
REFRESH1      = ^b1010;
REFRESH2      = ^b1011;
REFRESH3      = ^b1100;
REFRESH4      = ^b1101;
REFRESH5      = ^b1110;
REFRESH6      = ^b1111; "this state is never entered

"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"Holds state of A2 of the processor
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
state_diagram ODDACCESS
                state ASSERT:
                        if((CYCLE == IDLE) & A2) then DEASSERT
                else
                        ASSERT;

                state DEASSERT:
                        if((CYCLE == IDLE) & !A2) then ASSERT
                else
                        DEASSERT;
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"Even byte 0 CAS
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
state_diagram CASE_B0
                state ASSERT:
                        if(!Q3 & !CASEE) then DEASSERT
                else
                        if(Q3 & !CASEE) then DEASSERT
                else
                        ASSERT;

                state DEASSERT:
                        if(!Q3 & W_R & WAIT & CASEE & BE0) then ASSERT
                else
                        if(!Q3 & W_R & !WAIT & CASEE & BE0 & !BLAST) then ASSERT
                else
                        if(!Q3 & !W_R & CASEE & BE0) then ASSERT
                else
                        if(Q3 & CASEE) then ASSERT
                else
                        DEASSERT;
```

Table A-1. 40 MHz DRAM Controller PLD Equation (Sheet 5 of 19)

```
"Even byte 1 CAS
state_diagram CASE_B1
                state ASSERT:
                        if(!Q3 & !CASEE) then DEASSERT
                else
                        if(Q3 & !CASEE) then DEASSERT
                else
                        ASSERT;

                state DEASSERT:
                        if(!Q3 & W_R &  WAIT & CASEE & BE1) then ASSERT
                else
                        if(!Q3 & W_R & !WAIT & CASEE & BE1 & !BLAST) then ASSERT
                else
                        if(!Q3 & !W_R & CASEE & BE1) then ASSERT
                else
                        if(Q3 & CASEE) then ASSERT
                else
                        DEASSERT;
"Even byte 2 CAS
state_diagram CASE_B2
                state ASSERT:
                        if(!Q3 & !CASEE) then DEASSERT
                else
                        if(Q3 & !CASEE) then DEASSERT
                else
                        ASSERT;

                state DEASSERT:
                        if(!Q3 & W_R &  WAIT & CASEE & BE2) then ASSERT
                else
                        if(!Q3 & W_R & !WAIT & CASEE & BE2 & !BLAST) then ASSERT
                else
                        if(!Q3 & !W_R & CASEE & BE2) then ASSERT
                else
                        if(Q3 & CASEE) then ASSERT
                else
                        DEASSERT;
```

```
                state DEASSERT:
                        if(!Q3 & W_R &  WAIT & CASEE & BE3) then ASSERT
                else
                        if(!Q3 & W_R & !WAIT & CASEE & BE3 & !BLAST) then ASSERT
                else
                        if(!Q3 & !W_R & CASEE & BE3) then ASSERT
                else
                        if(Q3 & CASEE) then ASSERT
                else
                        DEASSERT;
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"Odd byte 0 CAS
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
state_diagram CAS0_B0
                state ASSERT:
                        if(!Q3 & !CAS0O) then DEASSERT
                else
                        if(Q3 & !CAS0O) then DEASSERT
                else
                        ASSERT;

                state DEASSERT:
                        if(!Q3 & W_R & WAIT & CAS0O & BE0 & BLAST & !LA2) then
ASSERT
                else
                        if(!Q3 & W_R & WAIT & CAS0O & BE0 & !BLAST & LA2) then
ASSERT
                else
                        if(!Q3 & !W_R & CAS0O & BE0) then ASSERT
                else
                        if(Q3 & CAS0O) then ASSERT
                else
                        DEASSERT;
```

Table A-1. 40 MHz DRAM Controller PLD Equation (Sheet 7 of 19)

```
"""""""
```

Table A-1. 40 MHz DRAM Controller PLD Equation (Sheet 8 of 19)

```
"""""""
```

Table A-1. 40 MHz DRAM Controller PLD Equation (Sheet 9 of 19)

```
"""""""
```

```
                else
                        if((CYCLE == REFRESH5)) then DEASSERT
                else
                        ASSERT;

                state DEASSERT:
                        if((CYCLE == IDLE) & ADS & !REFREQ & !ACC_PEND
                          & DRAMADDR & !SRASE) then ASSERT
                else
                        if((CYCLE == ACCESS0)) then ASSERT
                else
                        if((CYCLE ==  REFRESH2) & REFEVEN) then ASSERT
                else
                        DEASSERT;
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"Odd RAS
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
state_diagram RASODD
                state ASSERT:
                        if((CYCLE == ACCESS3) & !W_R & BLAST) then DEASSERT
                else
                        if((CYCLE == ACCESS4) & W_R & BLAST & LA2) then DEASSERT
                else
                        if((CYCLE == ACCESS5) & BLAST) then DEASSERT
                else
                        if((CYCLE == REFRESH5)) then DEASSERT
                else
                        ASSERT;

                state DEASSERT:
                        if((CYCLE == IDLE) & ADS & !REFREQ & !ACC_PEND
                          & DRAMADDR & !SRASE) then ASSERT
                else
                        if((CYCLE == ACCESS0))  then ASSERT
                else
                        if((CYCLE == REFRESH2) & !REFEVEN) then ASSERT
                else
                        DEASSERT;
```

Table A-1. 40 MHz DRAM Controller PLD Equation (Sheet 11 of 19)

```
"""""""
```

Table A-1. 40 MHz DRAM Controller PLD Equation (Sheet 12 of 19)

```
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"Even address counter
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
state_diagram A3EVEN
                state DEASSERT:
                        if((CYCLE == IDLE) &  A3) then ASSERT
                else
                        if((CYCLE == ACCESS4)) then ASSERT
                else
                        DEASSERT;

                state ASSERT:
                        if((CYCLE == IDLE) & !A3) then DEASSERT
                else
                        if((CYCLE == ACCESS4)) then DEASSERT
                else
                        ASSERT;
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"Odd address counter
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
state_diagram A3ODD
                state DEASSERT:
                        if((CYCLE == IDLE) &  A3) then ASSERT
                else
                        if((CYCLE == ACCESS5)) then ASSERT
                else
                        DEASSERT;

                state ASSERT:
                        if((CYCLE == IDLE) & !A3)  then DEASSERT
                else
                        if((CYCLE == ACCESS5)) then DEASSERT
                else
                        ASSERT;
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"Main DRAM state machine - ACCESS state machine
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
state_diagram CYCLE
                state IDLE:
                       if(ADS & !REFREQ & !ACC_PEND & DRAMADDR &  SRASE) then ACCESS0
                else
                       if(ADS & !REFREQ & !ACC_PEND & DRAMADDR & !SRASE) then ACCESS1
                else
                        if(!REFREQ & ACC_PEND) then ACCESS0
                else
                        if(REFREQ) then REFRESH0
                else
                        IDLE;

                state ACCESS0:
                        goto ACCESS1;

                state ACCESS1:
                        if(W_R & !LA2) then ACCESS3
                else
                        if(!W_R & !LA2) then ACCESS4
                else
                        ACCESS2;

                state ACCESS2:
                        goto ACCESS3;
```

```
                IDLE;

        state ACCESS6:
                goto IDLE;

        state ACCESS7:
                goto REFRESH0;

        state REFRESH0:
                goto REFRESH1;

        state REFRESH1:
                goto REFRESH2;

        state REFRESH2:
                goto REFRESH3;

        state REFRESH3:
                goto REFRESH4;

        state REFRESH4:
                goto REFRESH5;

        state REFRESH5:
                goto IDLE;

        state REFRESH6:
                goto IDLE;
"""""""
```

```
state Z2:
        if(Q3) then Z15
else
        Z1;

state Z3:
        if(Q3) then Z15
else
        Z2;

state Z4:
        if(Q3) then Z15
else
        Z3;

state Z5:
        if(Q3) then Z15
else
        Z4;

state Z6:
        if(Q3) then Z15
else
        Z5;

state Z7:
        if(Q3) then Z15
else
        Z6;

state Z8:
        if(Q3) then Z15
else
        Z7;

state Z9:
        if(Q3) then Z15
else
        Z8;

state Z10:
        if(Q3) then Z15
else
        Z9;
```

Table A-1. 40 MHz DRAM Controller PLD Equation (Sheet 15 of 19)

```
state Z11:
        if(Q3) then Z15
else
        Z10;

state Z12:
        if(Q3) then Z15
else
        Z11;

state Z13:
        if(Q3) then Z15
else
        Z12;

state Z14:
        if(Q3) then Z15
else
        Z13;

state Z15:
        if(Q3) then Z15
else
        Z14;
```

Table A-1. 40 MHz DRAM Controller PLD Equation (Sheet 16 of 19)

```
"Refresh Counter 2

state_diagram REFCT2
                state Z0:
                        if(Q3) then Z15
                else
                        Z0;

                state Z1:
                        if(Q3) then Z15
                else
                        if(!Q3 & (REFCT1 == Z0)) then Z0;
                else
                        Z1;

                state Z2:
                        if(Q3) then Z15
                else
                        if(!Q3 & (REFCT1 == Z0)) then Z1;
                else
                        Z2;

                state Z3:
                        if(Q3) then Z15
                else
                        if(!Q3 & (REFCT1 == Z0)) then Z2;
                else
                        Z3;

                state Z4:
                        if(Q3) then Z15
                else
                        if(!Q3 & (REFCT1 == Z0)) then Z3;
                else
                        Z4;

                state Z5:
                        if(Q3) then Z15
                else
                        if(!Q3 & (REFCT1 == Z0)) then Z4;
                else
                        Z5;

                state Z6:
                        if(Q3) then Z15
                else
                        if(!Q3 & (REFCT1 == Z0)) then Z5;
                else
                        Z6;

                state Z7:
                        if(Q3) then Z15
                else
                        if(!Q3 & (REFCT1 == Z0)) then Z6;
                else
                        Z7;
```

```
state Z10:
        if(Q3) then Z15
else
        if(!Q3 & (REFCT1 == Z0)) then Z9;
else
        Z10;

state Z11:
        if(Q3) then Z15
else
        if(!Q3 & (REFCT1 == Z0)) then Z10;
else
        Z11;

state Z12:
        if(Q3) then Z15
else
        if(!Q3 & (REFCT1 == Z0)) then Z11;
else
        Z12;

state Z13:
        if(Q3) then Z15
else
        if(!Q3 & (REFCT1 == Z0)) then Z12;
else
        Z13;

state Z14:
        if(Q3) then Z15
else
        if(!Q3 & (REFCT1 == Z0)) then Z13;
else
        Z14;

state Z15:
        if(Q3) then Z15
else
        if(!Q3 & (REFCT1 == Z0)) then Z14;
else
        Z15;
```

```
WAIT =     (CYCLE == ACCESS0) & W_R
         # (CYCLE == ACCESS1) & W_R
         # (CYCLE == ACCESS2) & W_R
         # (CYCLE == ACCESS3) & W_R;

[!MUX,!RASE,!RASO,!CASEE,!CASOO,A3E,A3O,!REFEVEN].clk = CLK1;
[!MUX,!RASE,!RASO,!CASEE,!CASOO,A3E,A3O,!REFEVEN].pr = RESET;

[!CASEB0,!CASEB1,!CASEB2,!CASEB3,!CASOB0,!CASOB1,!CASOB2,!CASOB3].clk = CLK1;

[T3,T2,T1,T0,S3,S2,S1,S0].clk = CLK1;
[T3,T2,T1,T0,S3,S2,S1,S0].pr = RESET;
"
"Refresh required indicator
"
REFREQ = !T3 & !T2 & !T1 & !T0 & !S3 & !S2 & !S1 & !S0;
"
"FLASH Chip Select
"
FLASHCS :=     ADS & FLASHADDR & !APK_ACTIVE
           # !ADS & !BLAST & FLASHCS;
"
"FLASH OE control
"
FLASHRD = FLASHCS & W_R;
"
"XCR OE control
"
XCROE :=    FLASHCS & !BLAST & !APK_ACTIVE
          # !ADS & LEDADDR & !BLAST;
"
"XCR DIR control
"
XCRDIR = W_R;
"
"Software reset indicator
"
SWRST :=     ADS & SWRSTADDR
         # !ADS & !BLAST & SWRST;
"
"Triggers the 7705
"
TRIGRST := TRIGRST;
TRIGRST.RE = SWRST;
```

Table A-1. 40 MHz DRAM Controller PLD Equation (Sheet 19 of 19)

```
"
"Pulse the HEX DISPLAY
"
LED_LAT  := !ADS & LEDADDR & XCROE & !BLAST;
"
"Latched RASE or RASO
"
SRASE := RASE # RASO;
"
"DRAM data path control while reading
"
RDEN = !Q3 &  W_R & RASE;
"
" Even DRAM data path control while writing
"
WRE  = !Q3 & !W_R & RASE;
"
"Odd DRAM data path control while writing
"
WRO  = !Q3 & !W_R & RASE;

[!FLASHCS,!XCROE,!SWRST,!TRIGRST,LED_LAT,SRASE].clk = CLK1;
[!FLASHCS,!XCROE,!SWRST,!TRIGRST].pr = RESET;
LED_LAT.RE = RESET;
"
"Latched external reset
"
RESET := EXTRST;
RESET.CLK = CLK1;

" Test vectors

end CX40T
```

Table A-2. Signal and Product Term Allocation

OUTPUTS		BURIED MACROCELLS	
Signal	Product Terms	Signal	Product Terms
$\overline{\text{RASE}}$	7	$\overline{\text{CASEE}}$	10
$\overline{\text{RASO}}$	7	$\overline{\text{CASOO}}$	9
$\overline{\text{READY}}$	6	Q3	3
A3E	3	Q2	8
A3O	3	Q1	9
$\overline{\text{SELA}}$	4	Q0	9
$\overline{\text{MUX}}$	1	$\overline{\text{ACC\_PEND}}$	5
$\overline{\text{RDEN}}$	1	$\overline{\text{LA2}}$	5
$\overline{\text{CASEB3}}$	3	$\overline{\text{WAIT}}$	3
$\overline{\text{CASEB2}}$	3	$\overline{\text{REFEVEN}}$	5
$\overline{\text{CASEB1}}$	3	T3	2
$\overline{\text{CASEB0}}$	3	T2	8
$\overline{\text{CASOB3}}$	5	T1	7
$\overline{\text{CASOB2}}$	5	T0	6
$\overline{\text{CASOB1}}$	5	S3	5
$\overline{\text{CASOB0}}$	5	S2	4
$\overline{\text{FLASHCS}}$	2	S1	3
$\overline{\text{FLASHRD}}$	1	S0	2
$\overline{\text{TRIGRST}}$	1	$\overline{\text{REFREQ}}$	1
$\overline{\text{XCROE}}$	2	$\overline{\text{SWRST}}$	2
$\overline{\text{XCRDIR}}$	1	$\overline{\text{SRASE}}$	1
$\overline{\text{RESET}}$	1		
$\overline{\text{WRE}}$	1		
$\overline{\text{WRO}}$	1		
LED_LAT	1		
$\overline{\text{SELB}}$	4		

intel

AP-712

APPLICATION NOTE

DRAM Controller for i960® JA/JF/JD Microprocessors

February 9, 1995

Order Number: 272674-001

DRAM Controller for i960® JA/JF/JD Microprocessors

February 9, 1995

Order Number: 272674-001

1.0 INTRODUCTION

This application note describes a DRAM controller for use with Intel's i960® JA/JF/JD microprocessors. Other application notes are available which describe DRAM controllers for the i960 CA and CF processors; see Section 6.0, RELATED INFORMATION for ordering information.

This DRAM controller's design features include:

- Non-interleaved design
- Can use standard 70 ns DRAM SIMM
- 3-1-1-1 wait state burst reads at 33 MHz
- 2-1-1-1 wait state burst writes at 33 MHz
- No delay lines

This application note contains some general DRAM controller theory as well as this design's state machine definitions and timing diagrams. It also contains the PLD equations which were used to build and test the prototype design. Timing analysis was verified with Timing Designer*. PLD equations were created in ABEL* as a device-independent design. Schematics were developed with OrCAD*. The timing analysis, schematics and PLD files are available through Intel's America's Application Support BBS, at (916) 356-3600.

1.1 Design Goals

A primary goal was to implement a single or dual bank 32-bit DRAM controller with the minimum number of components, using a standard 72-pin fast page mode DRAM SIMM. Such a design may be useful in embedded systems where space is at a premium. Accordingly, this design avoids such techniques as write posting, parity support and bank interleaving. A non-interleaved design significantly reduces system cost and complexity by using fewer components and logic. The memory in this design can be divided into two banks for addressing flexibility.

1.2 i960 Jx Processor Bus/Core Frequencies

This DRAM controller is designed for use with a 5 volt i960 Jx processor with bus frequencies up to 40 MHz. The design was verified with a 33 MHz version (production 40 MHz devices were not available at the time this document was published).

Available 80960 JA/JF speeds are 16 MHz, 25 MHz and 33 MHz. Intel's clock-doubled version, the 80960JD, operates at 16.7 MHz, 20 MHz and 25 MHz bus speeds, with the core operating at 33.3 MHz, 40 MHz and 50 MHz, respectively. Table 1 details the various DRAM speeds, bus speeds and wait state combinations that were tested with this design.

1.3 Page Mode DRAM SIMM

Page mode DRAM allows faster memory access by keeping the same row address while selecting random column addresses within that row. A new column address is selected by deasserting $\overline{CAS}$ while keeping $\overline{RAS}$ active and then asserting $\overline{CAS}$ with the new column address valid to the DRAM. Page mode operation works very well with burst buses, such as the i960 CA/CF processor bus, in which a single address cycle can be followed by up to four data cycles.

All $\overline{WE}$ pins on the SIMM are tied to a common $\overline{WE}$ line; this feature requires the use of early write cycles. In an early write cycle, write data is referenced to the falling edge of $\overline{CAS}$, not the falling edge of $\overline{WE}$.

The DRAM SIMM has four $\overline{CAS}$ lines, one for each eight (nine) bits in a 32-bit (36-bit) SIMM module. The four $\overline{CAS}$ lines control the writing to individual bytes within the SIMM. Each $\overline{CAS}$ signal is asserted with its respective $\overline{BE}$.

1.4 Burst Capabilities for 32-Bit Bus

The i960 Jx processor can access up to four data words per request. A request starts with the processor asserting $\overline{ADS}$ in the address cycle, and ends by the processor asserting $\overline{BLAST}$ in the last data cycle. Figure 1 shows $\overline{ADS}$ and $\overline{BLAST}$ timings for a quad-word write request.

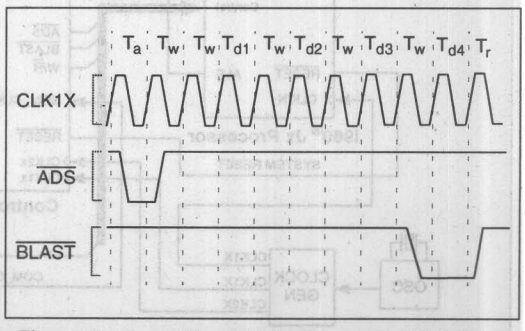


Figure 1. Quad-Word Write Request with 2,1,1,1 Wait State Profile

The processor's burst capabilities on a 32-bit bus include:

- Quad-word and triple-word requests start on quad-word boundaries (A3 = 0, A2 = 0).
- Double-word requests start on double-word boundaries (A3 = X, A2 = 0).
- Single-word requests can start on any word boundary (A3 = X, A2 = X).
- Any request starting on an odd word boundary never bursts (A3 = X, A2 = 1).

2.0 BASIC DRAM CONTROLLER

The main state machine of the DRAM controller in this design is clocked using CLKIN (CLK1X clock). The DRAM region is set at 0xAxxxxxxx for address decoding and uses the entire 256 Mbyte region. The region must also be configured as 32 bit in the processor's PMCON10_11 register. The controller is implemented as a four-bit state machine and is responsible for sequencing accesses as well as refreshes to the DRAM banks. Figure 2 shows the basic DRAM controller state machine.

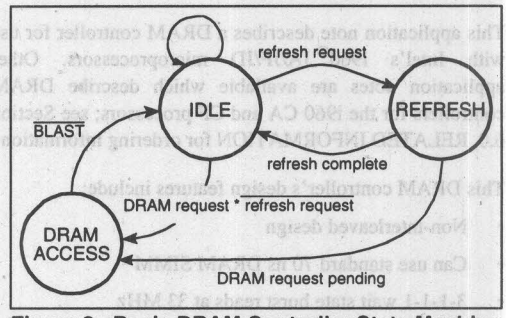


Figure 2. Basic DRAM Controller State Machine

The first state is IDLE. The state machine transitions from the IDLE state based on two events:

- DRAM refresh requests
- DRAM requests from the processor

The DRAM controller (Figure 3) has four distinct blocks: control logic, address latches, address muxes, and the DRAM SIMM. This section describes each block.

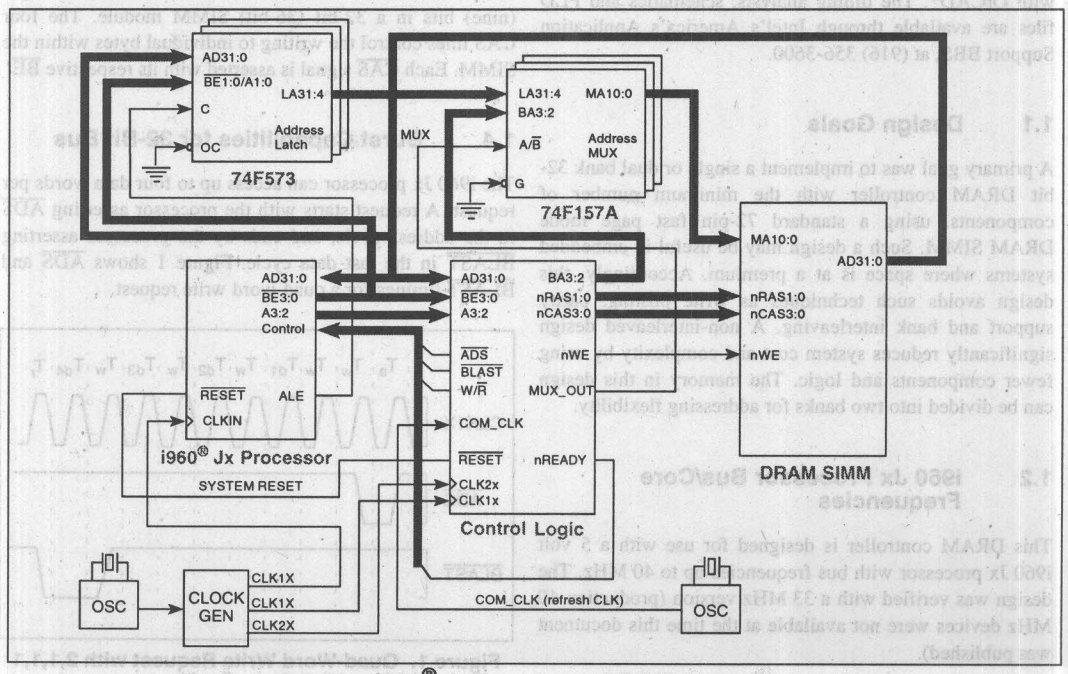


Figure 3. i960® Jx Processor DRAM Controller

2.1 Control Logic

The DRAM controller is centered around a four-bit state machine that controls accessing as well as refreshing of the DRAM banks. All signals are generated based on the four-bit state machine's outputs.

2.1.1 Refresh Logic

Typical 2 Mbyte x32 DRAMs need to be refreshed every 15.5 µs. The DRAM controller refreshes the banks using CAS-before-RAS refresh when the request is granted.

A refresh request has priority over a processor request. When a processor and a refresh request occur simultaneously, the DRAM controller sequences a refresh to the DRAM banks while the ACC_PENDING state machine posts the processor request. The pending request is then serviced after the refresh is done.

In the tested design, an eight-bit synchronous up-counter generates refresh requests. The counter is clocked using COM_CLK (7.3728 MHz clock). REFRESH_SYNC is asserted each time the counter reaches 98 (13.7 µs) which in turn asserts DRAM_REF. The REFRESH_SYNC equation synchronizes the output (when counter reaches 98) of the REFRESH state machine (which uses COM_CLK) to CLK1X. The refresh counter is reset to zero and counting resumes after the DRAM_STATES state machine completes servicing the refresh. During reset, the counter is also loaded with a zero.

2.1.2 Clock Generation

In this design a Motorola* MC88916D low skew CMOS PLL was used to generate the clock signals for the DRAM controller. The MC88916D uses the output of the oscillator as an input and outputs a 2X and 4X copy of the oscillator output:

- The 2X output is referred to as CLKIN or CLK1X
- The 4X output is referred to as CLK2X

The MC88916D produces a very low skew copy of both CLK1X and CLK2X. At 33 MHz, the maximum skew between the MC88916 and any of its outputs is ±1 ns, while the skew between any of the individual outputs is ±750 ps under equal loading conditions. In this design all clock lines are series terminated with 22-ohm resistors.

2.1.3 Wait State Profile

Table 1 provides typical wait state profiles for read and write accesses. the listed wait states have been tested with the design by changing the clock frequency and wait states and meeting the timing requirements for MUX, $\overline{RAS}$ and $\overline{CAS}$.

Table 1. i960® Jx Processor's Possible DRAM Wait State Combinations

DRAM Access Time	Mode	16 MHz Bus	20 MHz Bus	25 MHz Bus	33 MHz Bus	40 MHz Bus[1]
60ns	Read	1, 1, 1, 1	1, 1, 1, 1	1, 1, 1, 1	3, 1, 1, 1	3, 1, 1, 1
	Write	1, 0, 0, 0	1, 0, 0, 0	1, 1, 1, 1	2, 1, 1, 1	2, 1, 1, 1
70ns	Read	1, 1, 1, 1	1, 1, 1, 1	2, 1, 1, 1	3, 1, 1, 1	3, 2, 2, 2
	Write	1, 0, 0, 0	1, 0, 0, 0	1, 1, 1, 1	2, 1, 1, 1	2, 1, 1, 1
80ns	Read	1, 1, 1, 1	1, 1, 1, 1	2, 1, 1, 1	3, 1, 1, 1	4, 2, 2, 2
	Write	1, 0, 0, 0	1, 1, 1, 1	2, 1, 1, 1	3, 1, 1, 1	3, 2, 2, 2
100ns	Read	1, 1, 1, 1	2, 1, 1, 1	3, 1, 1, 1	4, 2, 2, 2	4, 3, 3, 3
	Write	1, 0, 0, 0	2, 1, 1, 1	3, 1, 1, 1	3, 1, 1, 1	4, 2, 2, 2

NOTE:
1. Contact your local Intel Sales Representative to determine availability of the 40 MHz i960 JA/JF processor.

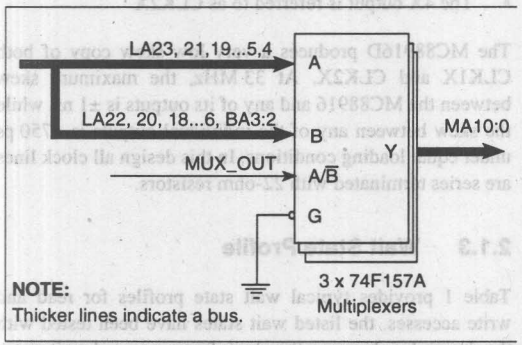


Figure 4. DRAM Address Multiplexers

2.3 Address Latch Path

As shown in Figure 5, the addresses are transferred to the latch outputs when the ALE signal is asserted. The latched addresses are always seen at the outputs by tying the output enables low.

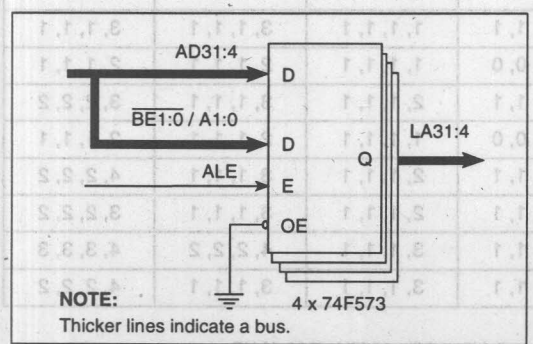


Figure 5. Address Latches

3.0 STATE MACHINES AND SIGNALS

This section describes the state machines and signals used in this design. Most of the state machines are simple and the PLD equations can be referenced in APPENDIX A. The DRAM controller's state machine DRAM_STATES is the most complex of all the state machines; for that reason, this application note provides more detail on the operations of this state machine.

In this design, some of the state machines are clocked with the CLK1X clock (bus clock frequency), while others are clocked off the CLK2X clock (twice the bus clock frequency), or the COM_CLK (communications\UART clock frequency).

Certain state machines and their outputs were clocked off of CLK2X so that their outputs could use PH1 or PH2 as conditions for transition. The high portion of CLK1X is referenced as PH1 (phase one). The low portion of CLK1X is referenced as PH2 (phase 2).

All PLD equations are written in ABEL. APPENDIX A, PLD EQUATIONS, contains a listing of the PLD equations file. The state machine transitions described here follow the ABEL conventions for logic operators.

- ! represents NOT, bit-wise negation
- & represents AND
- # represents OR

In this design signals do not have a polarity assigned, signals that begin with a "n" signify that the signal is active low. For example, nRAS0 refers to the non-asserted state (SET_HIGH) and !nRAS0 refers to the asserted state (SET_LOW).

3.1 LATCH_ACC State Machine

The LATCH_ACC state machine is a one-bit state machine that monitors DRAM requests from the processor. This state machine is necessary because a DRAM request may occur during a refresh. DRAM refresh has priority over a processor's request. Therefore, this state machine is used to post the processor request. This state machine generates the LATCHACC signal.

3.2 DRAM_ACC Signal

The DRAM_ACC signal monitors DRAM requests from the processor, and pending DRAM requests from LATCH_ACC state machine. When conditions are met DRAM_ACC initiates the DRAM portion of the DRAM controller. The DRAM_ACC state machine generates the DRAMACC signal.

3.3 DRAM_BANK State Machine

The DRAM_BANK state machine is a one-bit state machine that decodes AD20 to select between DRAM bank 0 (!AD20) or DRAM bank 1 (AD20). This state machine is clocked using the CLK1X clock. The DRAM_BANK state machine generates the DRAMBANK signal.

DRAM bank decoding is dependent on which size DRAM module is used. The following address lines provide contiguous memory:

- AD20 is decoded for a 2 Mbyte SIMM
- AD22 is decoded for a 8 Mbyte SIMM
- AD24 is decoded for a 32 Mbyte SIMM
- For 1 Mbyte, 4 Mbyte and 16 Mbyte SIMM modules, only one bank is possible

The DRAM_BANK state machine is currently fixed for 2 Mbyte SIMM. If a size other than this default size is needed, replace AD20 with the appropriate address bit.

3.4 MUX State Machine

The MUX state machine is a one-bit state machine that is used to control the address multiplexers, essentially to select between row (MUX) or column addresses (!MUX). It is clocked using the CLK2X clock. This state machine generates the MUX_OUT signal.

3.5 BURST_ADDR State Machine

The BURST_ADDR state machine is a two-bit state machine that is toggled on burst accesses to select the next data word (next column data). The state machine is initially loaded with the value of the processor's A3:2 during the address state (IDLE) and then increments after each data state (DATA). This state machine is clocked using the CLK1X clock, and generates the BA3:2 signal. These signals are an input to the DRAM Address Muxes.

3.6 REFRESH State Machine

The REFRESH state machine is a seven-bit state machine that is used as a counter to indicate when a refresh is required. The state machine is clocked using COM_CLK (7.3728 MHz). In the tested system, COM_CLK was present to drive a UART for communications. This state machine sequences from 0 to 98 then resets to zero. Sequencing begins when DRAM_REF is not asserted.

3.7 REFRESH_SYNC Signal

The REFRESH_SYNC signal monitors the REFRESH state machine counter. REFRESH_SYNC is asserted when the counter reaches a count of 98. The REFRESH_SYNC equation synchronizes the output (when counter reaches 98) of the REFRESH state machine, which uses COM_CLK, to CLK1X.

3.8 DRAM_REF State Machine

The DRAM_REF state machine is a one-bit state machine that uses the output of the REFRESH_SYNC equation. This state machine is clocked using the CLK1X clock, and generates the DRAMREF signal. DRAMREF signals the DRAM_STATES machine that a refresh is needed. DRAMREF is deasserted when the refresh states complete.

3.9 nPRE_CAS State Machine

The nPRE_CAS state machine is a one-bit state machine and generates the nPRECAS signal. This signal is generated one CLK2X clock cycle earlier than nCAS3:0. The output of this state machine is then fed to the nCAS3:0 equations where it is asserted at the correct time. This state machine is clocked using the CLK2X clock.

3.10 nCAS3:0 Signals

The nCAS3:0 signals are asserted when nPRE_CAS and the respective nBE3:0 signal are asserted. The output of this state machine is then fed to the DRAM. This equation is clocked using the CLK2X clock.

3.11 nRAS0 State Machine

The nRAS0 state machine is a one-bit state machine that is used to generate the $\overline{RAS}$ signal for bank 0. This state machine is clocked using the CLK2X clock, and generates the nRAS_0 signal.

3.12 nRAS1 State Machine

The nRAS1 state machine is a one-bit state machine that is used to generate the $\overline{RAS}$ signal for bank 1. This state machine is clocked using the CLK2X clock, and generates the nRAS_1 signal.

3.13 nREADY Signal

The nREADY signal is asserted during the data cycle of a DRAM access. Data is valid on the rising edge of CLK1X while nREADY is asserted. $\overline{BLAST}$ from the processor deasserts on the rising edge of CLK1X if nREADY is asserted during the last data cycle of the request. nREADY is clocked using the CLK1X clock.

3.14 nWE State Machine

The nWE state machine is a two-bit state machine that is asserted while a DRAM write is in progress. It controls the $\overline{WE}$ lines of the banks to perform early writes. This state machine is clocked using the CLK2X clock, and generates the nWE_OUT signal.

4.0 DRAM CONTROLLER ACCESS FLOW

This section explains how the DRAM_STATES state machine is sequenced while reading, writing, and refreshing the DRAM. Examples used are:

- single-word read and write access
- quad-word read and write access
- refresh

Figure 6 shows the complete DRAM_STATES state machine for the tested 33 MHz design. More detail on the states and state machine are given in the following examples.

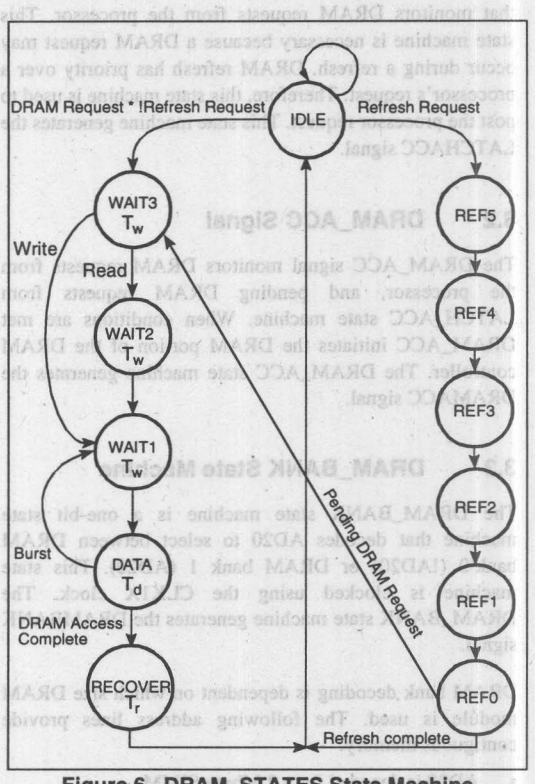


Figure 6. DRAM_STATES State Machine

4.1 Single-Word Access

Figure 7 shows the state diagram for a single-word read and write. Figure 8 is the single-word read timing diagram; Figure 9 is the single-word write timing diagram.

From the IDLE state the state machine waits for a DRAM access request or a refresh request. If a DRAM access is requested and not a refresh request, the state machine then transitions to the next state (WAIT3).

In WAIT3 state when PH1 is true, nRAS0 or nRAS1 is asserted depending on the bank selection; MUX is then asserted when PH2 is true. If the current access is a read, the machine proceeds to the WAIT2 state. If the current access is a write, the signals that normally assert in WAIT2 state (described below) now assert in WAIT3 state. The machine

proceeds to WAIT1 state bypassing WAIT2 state when the access is a write.

In WAIT2 state when PH2 is true, nCASx (x=3, 2, 1, or 0) is asserted if nPRE_CAS and the respective byte enable signal from the processor is asserted. The nWE signal is also asserted in WAIT2 state when PH2 is true. The machine then enters WAIT1 state.

When WAIT1 state is true, the respective nCAS signal is asserted when WAIT1 and PH2 are true. At this point, the processor has already asserted $\overline{\text{BLAST}}$. The machine then proceeds to the DATA state.

Data is valid in the DATA state (T_d) on the rising edge of CLK1X. When the DATA state is true, nREADY asserts. The respective nCAS signals deassert when DATA and PH2 are true. From the DATA state, the machine proceeds to the RECOVER state since $\overline{\text{BLAST}}$ is asserted.

When RECOVER is true, nREADY deasserts. When RECOVER and PH1 are true, MUX and the respective nRAS signals deassert completing the single word access.

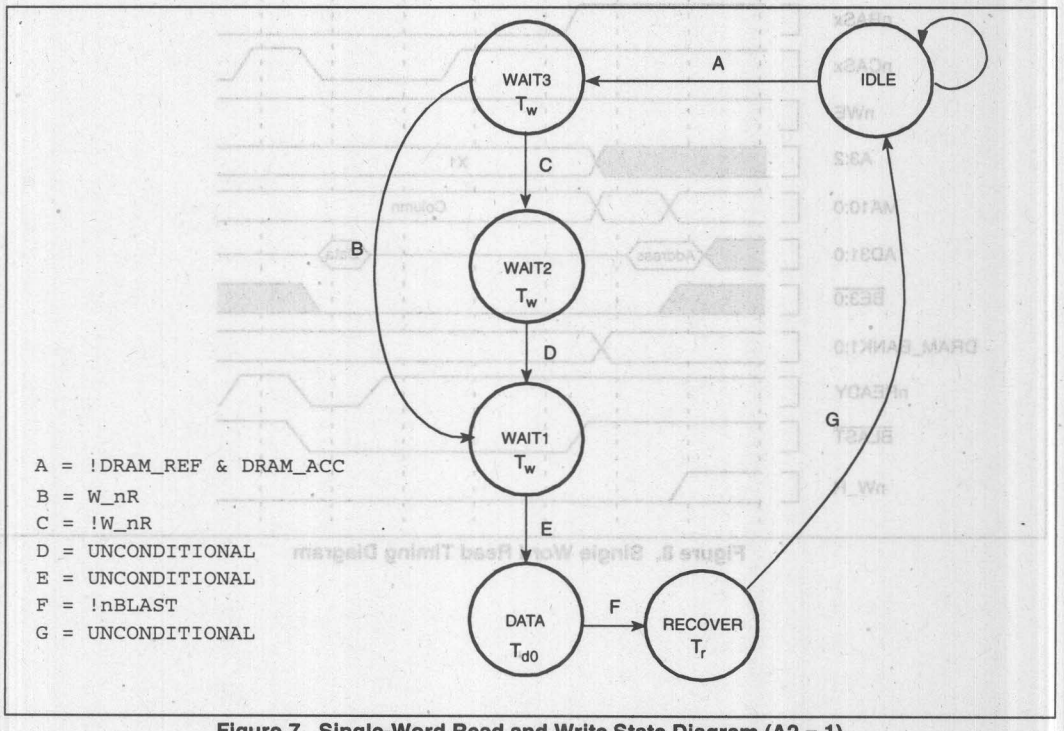


Figure 7. Single-Word Read and Write State Diagram (A2 = 1)

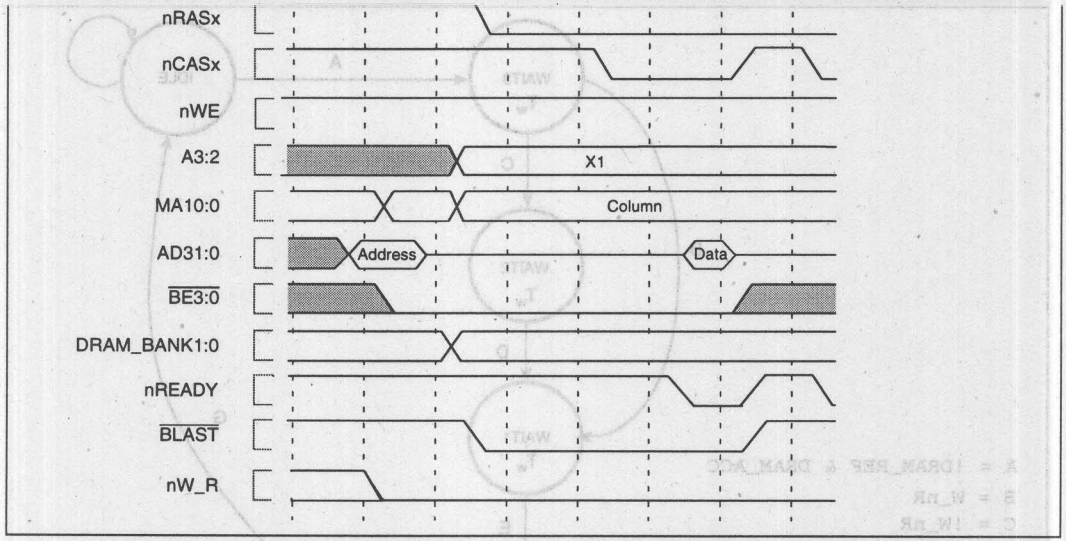


Figure 8. Single Word Read Timing Diagram

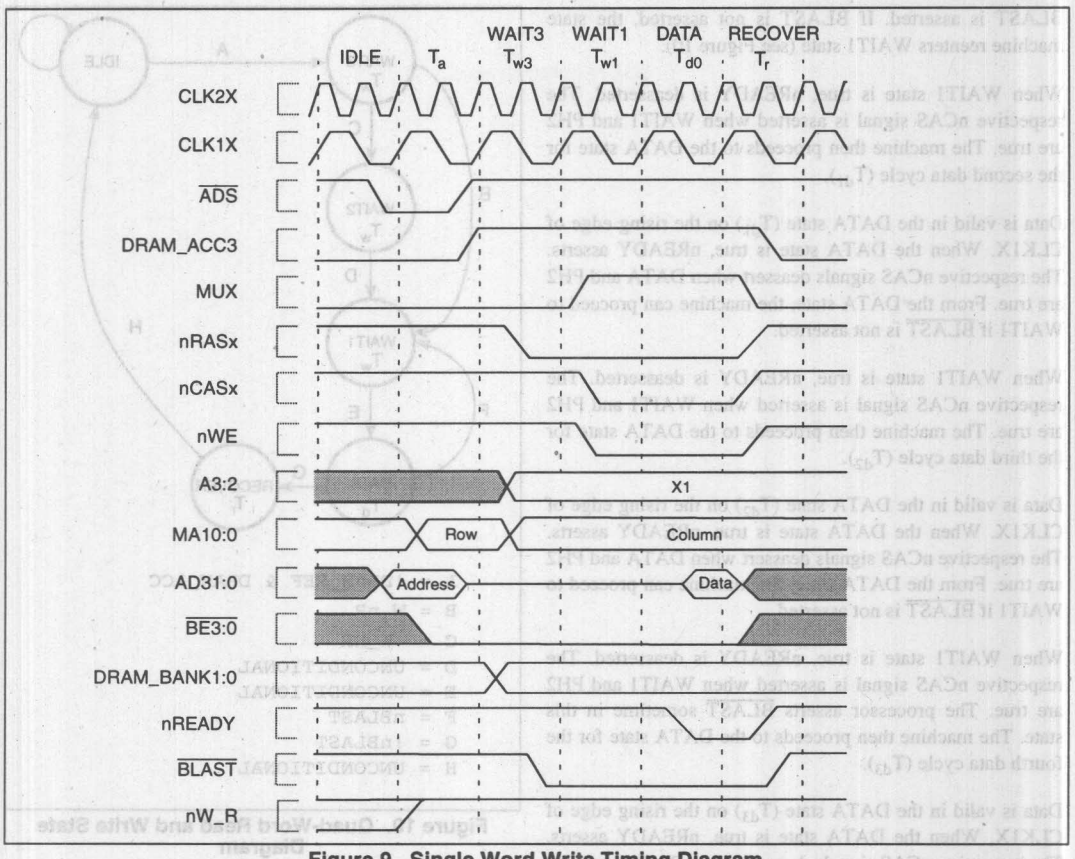


Figure 9. Single Word Write Timing Diagram

4.2 Quad-Word Access

Figure 10 shows the state diagram for a quad-word access. This state diagram also applies for triple-, and double-word accesses. Figure 11 is the quad-word read timing diagram; Figure 12 is the quad-word write timing diagram.

From the IDLE state the state machine waits for a DRAM access request or a refresh request. If a DRAM access is requested and not a refresh request, the state machine then transitions to the next state (WAIT3).

In WAIT3 state when PH1 is true, nRAS0 or nRAS1 is asserted depending on the bank selection; MUX is then asserted when PH2 is true. If the current access is a read, the machine proceeds to the WAIT2 state. If the current access is a write, the signals that normally assert in WAIT2 state (described below) assert in WAIT3 state. The machine proceeds to WAIT1 bypassing WAIT2 state when the access is a write (see Figure 10).

In WAIT2 state when PH2 is true, nCASx (x=3, 2, 1, or 0) is asserted if nPRE_CAS and the respective byte enable signal from the processor is asserted. The nWE signal is also asserted in WAIT2 state when PH2 is true. The machine then enters WAIT1 state.

From the WAIT1 state the machine proceeds to the DATA state.

The DATA state is the data cycle (T_d) of the access. Data is valid on the rising edge of CLK1X. When the DATA state is true, nREADY asserts. The respective nCAS signals deassert when DATA and PH2 are true. From the DATA

state, the machine can proceed to the RECOVER state if $\overline{BLAST}$ is asserted. If $\overline{BLAST}$ is not asserted, the state machine reenters WAIT1 state (see Figure 10).

When WAIT1 state is true, nREADY is deasserted. The respective nCAS signal is asserted when WAIT1 and PH2 are true. The machine then proceeds to the DATA state for the second data cycle (T_{d1}).

Data is valid in the DATA state (T_{d1}) on the rising edge of CLK1X. When the DATA state is true, nREADY asserts. The respective nCAS signals deassert when DATA and PH2 are true. From the DATA state, the machine can proceed to WAIT1 if $\overline{BLAST}$ is not asserted.

When WAIT1 state is true, nREADY is deasserted. The respective nCAS signal is asserted when WAIT1 and PH2 are true. The machine then proceeds to the DATA state for the third data cycle (T_{d2}).

Data is valid in the DATA state (T_{d2}) on the rising edge of CLK1X. When the DATA state is true, nREADY asserts. The respective nCAS signals deassert when DATA and PH2 are true. From the DATA state, the machine can proceed to WAIT1 if $\overline{BLAST}$ is not asserted.

When WAIT1 state is true, nREADY is deasserted. The respective nCAS signal is asserted when WAIT1 and PH2 are true. The processor asserts $\overline{BLAST}$ sometime in this state. The machine then proceeds to the DATA state for the fourth data cycle (T_{d3}).

Data is valid in the DATA state (T_{d3}) on the rising edge of CLK1X. When the DATA state is true, nREADY asserts. The respective nCAS signals deassert when DATA and PH2 are true. From the DATA state, the machine proceeds to the RECOVER state since $\overline{BLAST}$ is asserted.

When RECOVER is true, nREADY deasserts. When RECOVER and PH1 are true, MUX and the respective nRAS signals deassert, completing the quad word access.

The RECOVER state is a required state by the i960 Jx processor. This state is a function of the processor's $\overline{RDYRCV}$ pin. The processor continues to insert recovery states until it samples the $\overline{RDYRCV}$ pin HIGH. This function allows slow external devices more time to float their buffers before the processor drives address again. Transition to the next state, IDLE, is unconditional.

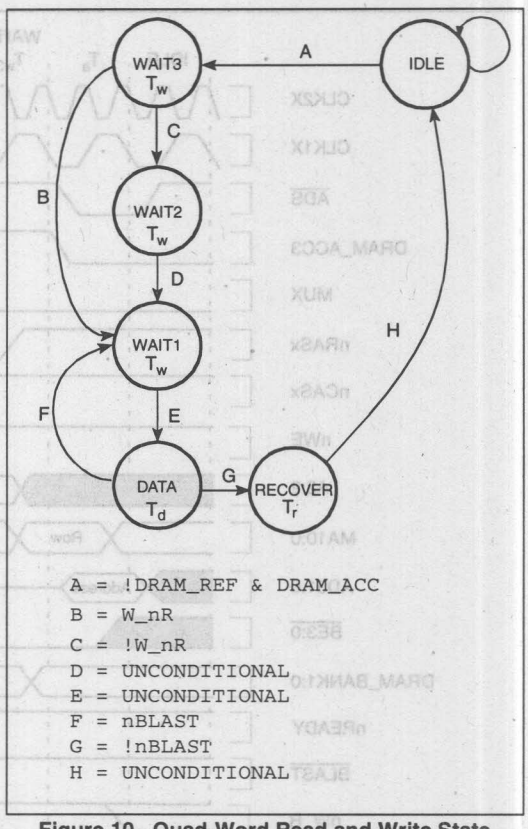


Figure 10. Quad-Word Read and Write State Diagram

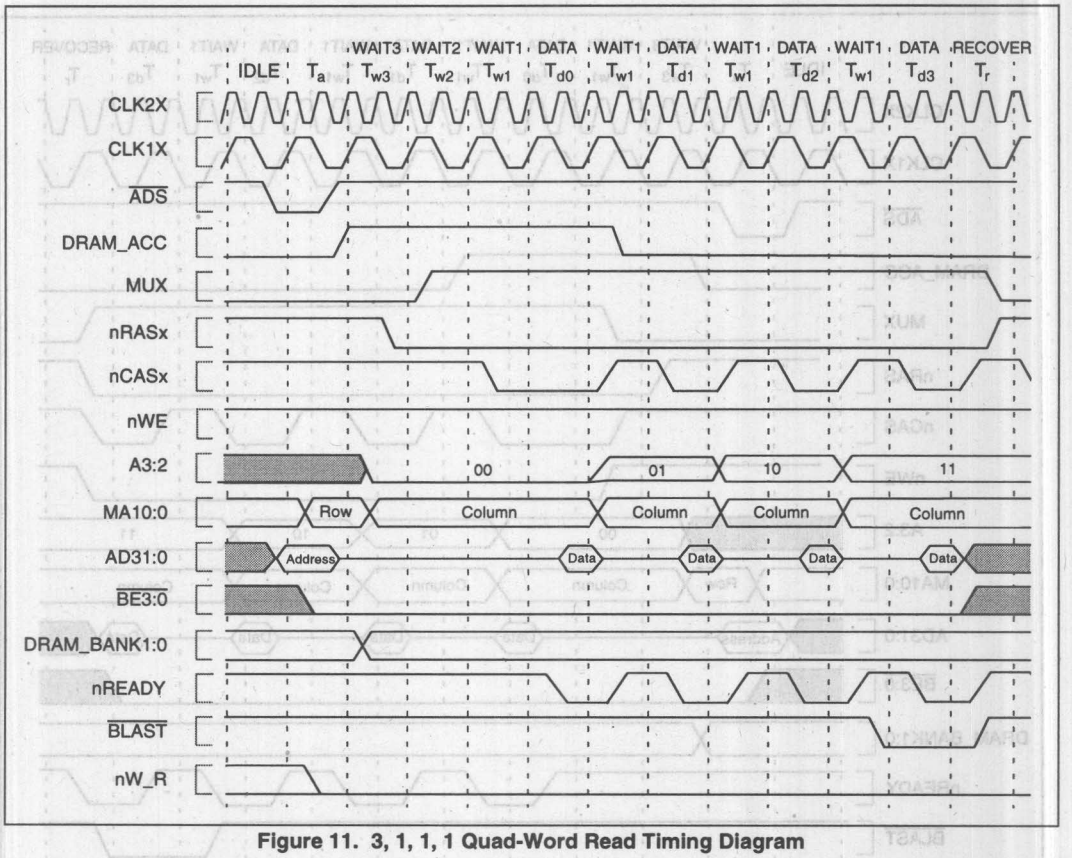


Figure 11. 3, 1, 1, 1 Quad-Word Read Timing Diagram

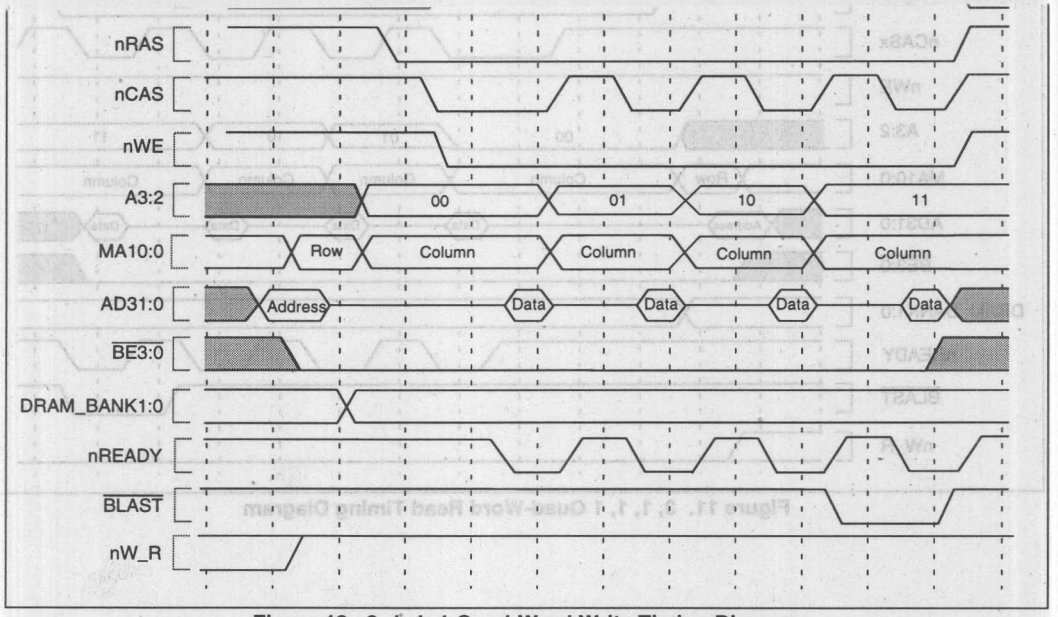


Figure 12. 2, 1, 1, 1 Quad-Word Write Timing Diagram

4.3 Refresh Cycles

In the tested design it was convenient to divide the communications\UART clock by 98 to provide a refresh request every 13.3 µs. The DRAM_STATES state machine services the request and sequences the refresh states. The example below describes the refresh states in the DRAM_STATES state machine.

Figure 13 shows the refresh state diagram; Figure 14 shows the refresh timing diagram with a pending DRAM request; Figure 15 shows the refresh timing diagram with no pending DRAM request.

In the IDLE state — if DRAMREF (see Section 3.8, DRAM_REF State Machine) is asserted and PH1 is true — all nCAS3:0 signals are asserted. Since DRAM_REF is asserted, the machine enters the REF5 state. The machine unconditionally proceeds to the REF4 state.

In the REF4 state when PH1 is true, the nRAS1:0 signals are asserted. The machine unconditionally proceeds to the REF3 state.

In the REF3 state when PH1 is true, the nCAS3:0 signals are deasserted. The machine unconditionally proceeds to the REF2 state.

In the REF2 state when PH2 is true, the nRAS1:0 signals are deasserted. The machine unconditionally proceeds to the REF1 state.

There are no signals that transition in state REF1. The machine unconditionally proceeds to the REF0 state.

When the REF0 state is true, the DRAMREF signal is deasserted. The machine then checks the DRAMACC status. If asserted, a DRAM access is pending and the machine proceeds to WAIT3 state (see Figure 13 and Figure 15). If DRAMACC is not asserted, the machine proceeds to the IDLE state (see Figure 13 and Figure 14).

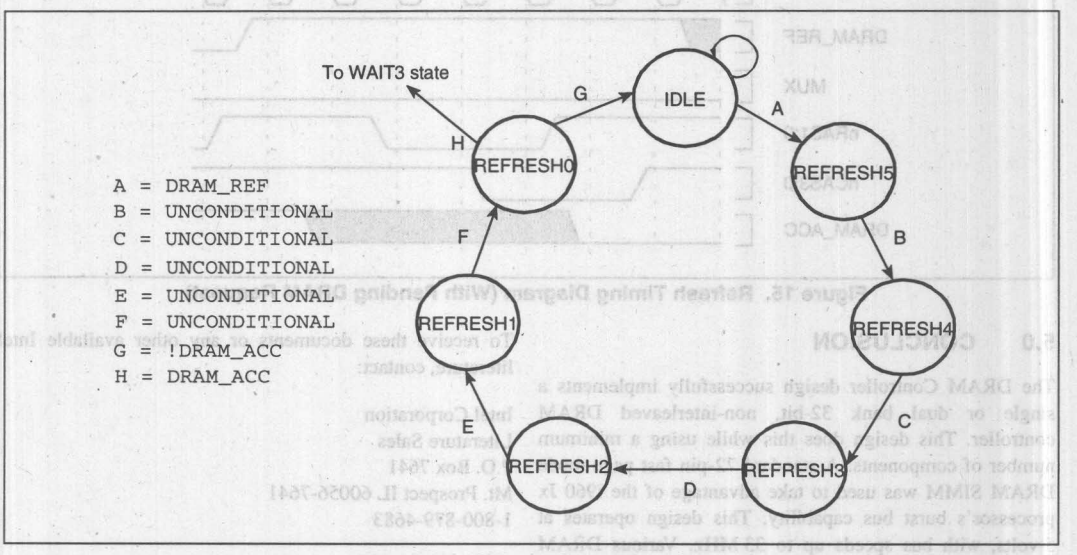


Figure 13. Refresh State Diagram

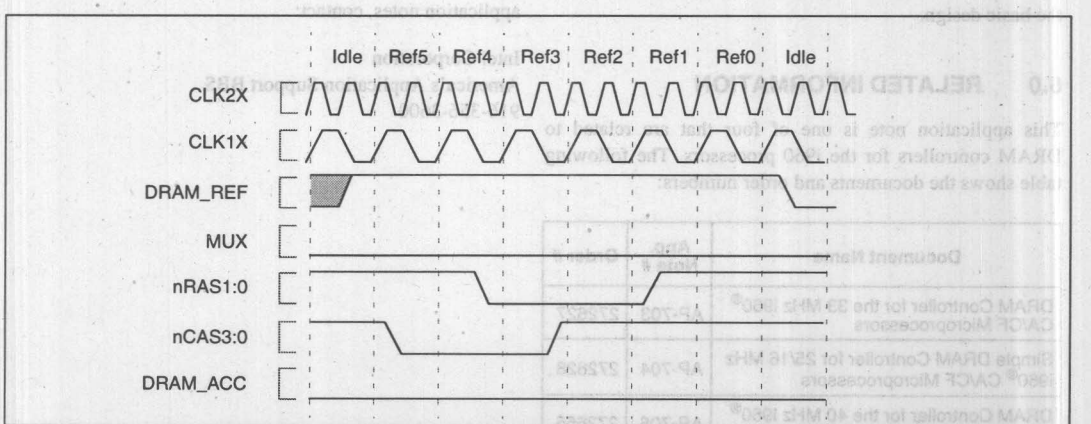


Figure 14. Refresh Timing Diagram (No Pending DRAM Request)

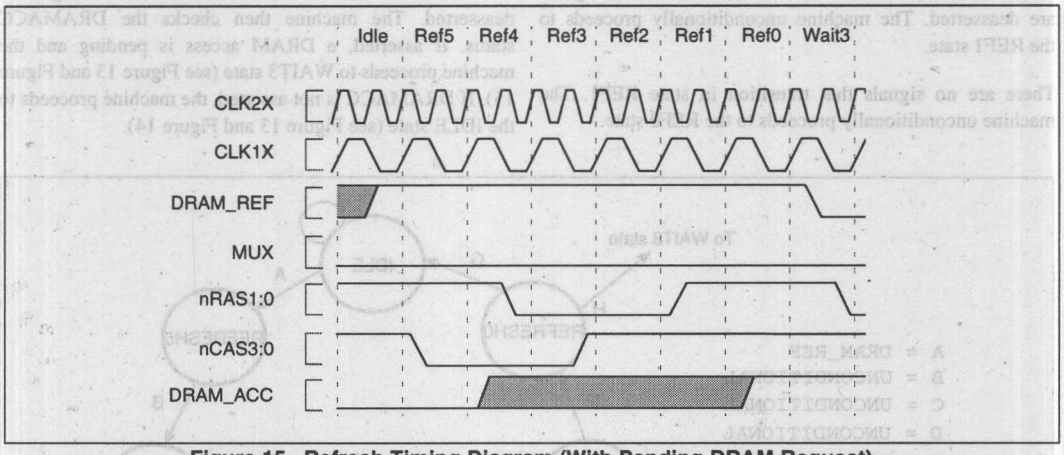


Figure 15. Refresh Timing Diagram (With Pending DRAM Request)

5.0 CONCLUSION

The DRAM Controller design successfully implements a single or dual bank 32-bit, non-interleaved DRAM controller. This design does this while using a minimum number of components. A standard 72-pin fast page mode DRAM SIMM was used to take advantage of the i960 Jx processor's burst bus capability. This design operates at 5 volts, with bus speeds up to 33 MHz. Various DRAM speeds, bus speeds up to 40 MHz, and wait state combinations were also successfully tested with minor changes to the basic design.

6.0 RELATED INFORMATION

This application note is one of four that are related to DRAM controllers for the i960 processors. The following table shows the documents and order numbers:

Document Name	App. Note #	Order #
DRAM Controller for the 33 MHz i960® CA/CF Microprocessors	AP-703	272627
Simple DRAM Controller for 25/16 MHz i960® CA/CF Microprocessors	AP-704	272628
DRAM Controller for the 40 MHz i960® CF Microprocessors	AP-706	272655

To receive these documents or any other available Intel literature, contact:

Intel Corporation
Literature Sales
P.O. Box 7641
Mt. Prospect IL 60056-7641
1-800-879-4683

To receive files that contain the timing analysis, schematics and PLD equations for this and the other DRAM controller application notes, contact:

Intel Corporation
America's Application Support BBS
916-356-3600

APPENDIX A
ABEL FILE

Table A-1 contains the PLD equations which were used to build and test the prototype design. Table A-2 defines signal and product term allocation. The PLD equations were created in ABEL* as a device-independent design. Using the ABEL software, a PDS file was created and then imported into PLDSHELL* software. PLDSHELL was used to "fit" the design into an Altera EPX780 FLEXlogic* PLD. PLDSHELL was also used to create the JEDEC file and to simulate the design.

Table A-1. i960® Jx Processor DRAM Controller ABEL File (Sheet 1 of 15)

```
Module          JXDRAM
Title           ' 80960 Jx DRAM Controller
Source File     JXDRAM.ABL
Revision        1.0

Date            11/15/94
Designer        Paul Durazo
                Intel 80960 Applications engineer'
" ********************************************************************************
" Non interleaved DRAM controller for the 80960JA/JF/JD with bus speeds up to 40MHz.
" One or two banks may be used depending on the DRAM size.
" See DRAM_BANK state machine for further information.
" DRAM start address is 0xA0000000
" DRAM wait state configuration is 3,1,1,1
" Active Low signals are designated with a (n) in front of the signal name.
" A (!) in front of a signal signifies it is cleared to 0.
" ie. nCAS is active low but not asserted, !nCAS is asserted or cleared to 0
" ********************************************************************************
"       Uxx     device  'iFX780_132';
" input signals
CLK1X           PIN; " 1X clock
CLK2X           PIN; " 2X clock
ICLK1X          PIN; " 1X clock fed to macrocell
COM_CLK         PIN; " Refresh  clock

AD31            PIN;
AD30            PIN;
AD29            PIN;
AD28            PIN;
AD22            PIN;
nADS            PIN;
W_nR            PIN;
nRESET          PIN;
nBLAST          PIN;
nBE3            PIN;
nBE2            PIN;
nBE1            PIN;
nBE0            PIN;
A3              PIN;
A2              PIN;
```

Table A-1. i960® Jx Processor DRAM Controller ABEL File (Sheet 2 of 15)

```
"output and node signals
nRAS_1          PIN     istype 'reg' ;
nRAS_0          PIN     istype 'reg' ;
nWE_OUT         PIN     istype 'reg' ;
MUX_OUT         PIN     istype 'reg' ;
BA3             PIN     istype 'reg' ;
BA2             PIN     istype 'reg' ;
DRAMREF         NODE    istype 'reg' ;
LATCH_ACC       NODE    istype 'reg' ;
DRAM_ACC        NODE    istype 'com' ;
DRAMBANK        NODE    istype 'reg' ;
REFRESH_SYNC    NODE    istype 'reg' ;
nREADY          PIN     istype 'reg' ;
nPRECAS         NODE    istype 'reg' ;
nCAS_3          PIN     istype 'reg' ;
nCAS_2          PIN     istype 'reg' ;
nCAS_1          PIN     istype 'reg' ;
nCAS_0          PIN     istype 'reg' ;
Q3              NODE    istype 'reg' ;
Q2              NODE    istype 'reg' ;
Q1              NODE    istype 'reg' ;
Q0              NODE    istype 'reg' ;
R6              NODE    istype 'reg' ; "Refresh counter bit 6
R5              NODE    istype 'reg' ; "Refresh counter bit 5
R4              NODE    istype 'reg' ; "Refresh counter bit 4
R3              NODE    istype 'reg' ; "Refresh counter bit 3
R2              NODE    istype 'reg' ; "Refresh counter bit 2
R1              NODE    istype 'reg' ; "Refresh counter bit 1
R0              NODE    istype 'reg' ; "Refresh counter bit 0

C               = .C.;
X               = .X.;

SET_LOW    = ^b0;
SET_HI     = ^b1;

" Sate       |  I/O pin
" references |  references
PH1             = [ICLK1X];             " high phase of 1X clock
PH2             = [!ICLK1X];            " low phase of 1X clock
DRAM_STATES     = [Q3, Q2, Q1, Q0];     " main dram state machine
nRAS0           = [nRAS_0];             " Bank 0 RAS signal
nRAS1           = [nRAS_1];             " Bank 1 RAS signal
nCAS3           = [nCAS_3];
nCAS2           = [nCAS_2];
nCAS1           = [nCAS_1];
nCAS0           = [nCAS_0];
MUX             = [MUX_OUT];
BURST_ADDR      = [BA3,BA2];
```

Table A-1. i960® Jx Processor DRAM Controller ABEL File (Sheet 3 of 15)

```
nWE             = [nWE_OUT];
DRAM_BANK       = [DRAMBANK];
DRAM_REF        = [DRAMREF];
LATCHACC        = [LATCH_ACC];
DRAMACC         = [DRAM_ACC];
REFRESH         = [R6, R5, R4, R3, R2, R1, R0];
nPRE_CAS        = [nPRECAS];

RECOVER = ^b0000;       " RECOVER is a recovery state and is required by the processor.
IDLE    = ^b0001;       " IDLE is the address state; determines if DRAM access has occurred.
WAIT3   = ^b0010;       " WAIT3-1 are wait states
WAIT2   = ^b0011;       "
WAIT1   = ^b0100;       "
DATA    = ^b0101;       " Data state
REF5    = ^b1000;       " REF5-1 are Refresh states
REF4    = ^b1001;       "
REF3    = ^b1010;       "
REF2    = ^b1011;       "
REF1    = ^b1100;       "
REF0    = ^b1101;       "

SA0 = ^b00;             " Burst address counter states
SA1 = ^b01;
SA2 = ^b10;
SA3 = ^b11;

RF0   = ^b0000000;      " Refresh counter States
RF1   = ^b0000001;
RF2   = ^b0000010;
RF3   = ^b0000011;
RF4   = ^b0000100;
RF5   = ^b0000101;
RF6   = ^b0000110;
RF7   = ^b0000111;
RF8   = ^b0001000;
RF9   = ^b0001001;
RF10  = ^b0001010;
RF11  = ^b0001011;
RF12  = ^b0001100;
RF13  = ^b0001101;
RF14  = ^b0001110;
RF15  = ^b0001111;
RF16  = ^b0010000;
RF17  = ^b0010001;
RF18  = ^b0010010;
RF19  = ^b0010011;
RF20  = ^b0010100;
RF21  = ^b0010101;
RF22  = ^b0010110;
RF23  = ^b0010111;
```

Table A-1. i960® Jx Processor DRAM Controller ABEL File (Sheet 4 of 15)

```
RF24  = ^b0011000;
RF25  = ^b0011001;
RF26  = ^b0011010;
RF27  = ^b0011011;
RF28  = ^b0011100;
RF29  = ^b0011101;
RF30  = ^b0011110;
RF31  = ^b0011111;
RF32  = ^b0100000;
RF33  = ^b0100001;
RF34  = ^b0100010;
RF35  = ^b0100011;
RF36  = ^b0100100;
RF37  = ^b0100101;
RF38  = ^b0100110;
RF39  = ^b0100111;
RF40  = ^b0101000;
RF41  = ^b0101001;
RF42  = ^b0101010;
RF43  = ^b0101011;
RF44  = ^b0101100;
RF45  = ^b0101101;
RF46  = ^b0101110;
RF47  = ^b0101111;
RF48  = ^b0110000;
RF49  = ^b0110001;
RF50  = ^b0110010;
RF51  = ^b0110011;
RF52  = ^b0110100;
RF53  = ^b0110101;
RF54  = ^b0110110;
RF55  = ^b0110111;
RF56  = ^b0111000;
RF57  = ^b0111001;
RF58  = ^b0111010;
RF59  = ^b0111011;
RF60  = ^b0111100;
RF61  = ^b0111101;
RF62  = ^b0111110;
RF63  = ^b0111111;
RF64  = ^b1000000;
RF65  = ^b1000001;
RF66  = ^b1000010;
RF67  = ^b1000011;
RF68  = ^b1000100;
RF69  = ^b1000101;
RF70  = ^b1000110;
RF71  = ^b1000111;
RF72  = ^b1001000;
```

Table A-1. i960® Jx Processor DRAM Controller ABEL File (Sheet 5 of 15)

```
RF73  = ^b1001001;
RF74  = ^b1001010;
RF75  = ^b1001011;
RF76  = ^b1001100;
RF77  = ^b1001101;
RF78  = ^b1001110;
RF79  = ^b1001111;
RF80  = ^b1010000;
RF81  = ^b1010001;
RF82  = ^b1010010;
RF83  = ^b1010011;
RF84  = ^b1010100;
RF85  = ^b1010101;
RF86  = ^b1010110;
RF87  = ^b1010111;
RF88  = ^b1011000;
RF89  = ^b1011001;
RF90  = ^b1011010;
RF91  = ^b1011011;
RF92  = ^b1011100;
RF93  = ^b1011101;
RF94  = ^b1011110;
RF95  = ^b1011111;
RF96  = ^b1100000;
RF97  = ^b1100001;
RF98  = ^b1100010;
RF99  = ^b1100011;
RF100 = ^b1100100;


"*****************************************************************************
"               DRAM STATE ASSIGNMENTS
"*****************************************************************************

state_diagram DRAM_STATES

state IDLE:                     " wait for a DRAM access or refresh request
       if(DRAM_REF) then REF5   " If refresh true begin refresh states
  else
     if(DRAMACC & !DRAM_REF) then WAIT3 " If DRAM access true continue to next state
  else
       IDLE;                    " Stay in IDLE state until refresh or DRAM access occurs

state WAIT3:                    " Tw, wait state 3
     if(W_nR) then WAIT1        " If a write skip the next state for 2,1,1,1 profile
  else
       WAIT2;                   " was a read so go to WAIT2 for 3,1,1,1 profile

state WAIT2:                    " Tw, wait state 2
       goto WAIT1;
```

```
        goto REF4;              " Begin Dram Refresh, nCAS asserted coming into this state
state REF4:
        goto REF3;              " Refresh State, nRAS asserted in phase 1
state REF3:
        goto REF2;              " Refresh State, nCAS deasserted in phase 1
state REF2:
        goto REF1;              " Refresh State, nRAS deasserted in phase 2
state REF1:
        goto REF0;              "
state REF0:
        if(DRAMACC) then WAIT3  " Dram Access occured during refresh go to WAIT3 state
  else
        goto IDLE;              " No Dram Access pending go to IDLE state

"*****************************************************************************
"nRAS0 F/F, looks at Bank 0
"*****************************************************************************
state_diagram nRAS0

        state SET_LOW:

                if((DRAM_STATES == RECOVER) & PH1) then SET_HI
        else
                if((DRAM_STATES == REF2) & PH2) then SET_HI

        else
                SET_LOW;

        state SET_HI:

                if((DRAM_STATES == REF4) & PH1) then SET_LOW
        else
                if((DRAM_STATES == WAIT3) & PH1 & !DRAM_BANK) then SET_LOW

        else
                SET_HI;
```

Table A-1. i960® Jx Processor DRAM Controller ABEL File (Sheet 7 of 15)

```
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"RAS1 F/F, looks at Bank 1
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
state_diagram nRAS1

        state SET_LOW:

                if((DRAM_STATES == RECOVER) & PH1) then SET_HI
        else
                if((DRAM_STATES == REF2) & PH2) then SET_HI

        else
                SET_LOW;

        state SET_HI:

                if((DRAM_STATES == REF4) & PH1) then SET_LOW
        else
                if((DRAM_STATES == WAIT3) & PH1 & DRAM_BANK) then SET_LOW

        else
                SET_HI;

"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
" MUX F/F
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
state_diagram MUX
        state SET_HI:
                if((DRAM_STATES == RECOVER) & PH1) then SET_LOW  " deassert MUX
        else
                SET_HI;

        state SET_LOW:
                if((DRAM_STATES == WAIT3) & PH2) then SET_HI      " assert MUX
        else
                SET_LOW;
```

Table A-1. i960® Jx Processor DRAM Controller ABEL File (Sheet 8 of 15)

```
"""""""
```

```
                SET_HI;

"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
" DRAM_BANK specifies which bank will be used (BANK 0 = !AD20) or (BANK 1 = AD20).
" For 2Mb simm decode AD20, 8Mb SIMM decode AD22, 32 Mb SIMM decode AD24.
" All other SIMM sizes only one bank is possible and nRAS1:
" 0 will have to be modified to reflect this.
" The default used in this design is a 2Mb SIMM which decodes AD20.
" If using a different SIMM size replace AD20 with the respective address bit
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
state_diagram DRAM_BANK
        state SET_HI:
                if(!nADS & AD31 & !AD30 & AD29 & !AD28 & !AD20)then SET_LOW "Bank0 sel
        else
                SET_HI;

        state SET_LOW:
                if(!nADS & AD31 & !AD30 & AD29 & !AD28 & AD20) then SET_HI "Bank1 sel
        else
                SET_LOW;

"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
" DRAM Refresh Request F/F
" When refresh counter = 98, REFRESH_SYNC is set. DRAM_REF is cleared upon leaving
" the refresh states of DRAM_STATES state machine
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
state_diagram DRAM_REF
        state SET_HI:
                if(DRAM_STATES == REF0) then SET_LOW    " Refresh states are complete
        else
                SET_HI;

        state SET_LOW:
                if(REFRESH_SYNC) then SET_HI            " Refresh required
        else
                SET_LOW;
```

Table A-1. i960® Jx Processor DRAM Controller ABEL File (Sheet 10 of 15)

```
"*************************************************************************
" Latch DRAM access
"*************************************************************************
state_diagram LATCH_ACC
        state SET_HI:
                if(DRAM_STATES == DATA) then SET_LOW          "in DRAM state machine deassert
        else
                SET_HI;

        state SET_LOW:
                if(!nADS & AD31 & !AD30 & AD29 & !AD28) then SET_HI "DRAM access occurred
        else
                SET_LOW;

"*************************************************************************
" DRAM Refresh counter
" The counter uses a 7.3728 MHz clock (COM_CLK). After 98 clocks (13.3 us) a refresh
" request (REFRESH_SYNC) bit is set.
"*************************************************************************
state_diagram REFRESH
STATE RF0:
        if(!DRAMREF) then RF1
else
        RF0;
STATE RF1:
        goto RF2;
STATE RF2:
        goto RF3;
STATE RF3:
        goto RF4;
STATE RF4:
        goto RF5;
STATE RF5:
        goto RF6;
STATE RF6:
        goto RF7;
STATE RF7:
        goto RF8;
STATE RF8:
        goto RF9;
STATE RF9:
        goto RF10;
STATE RF10:
        goto RF11;
STATE RF11:
        goto RF12;
STATE RF12:
        goto RF13;
```

Table A-1. i960® Jx Processor DRAM Controller ABEL File (Sheet 11 of 15)

```
STATE RF13:
        goto RF14;
STATE RF14:
        goto RF15;
STATE RF15:
        goto RF16;
STATE RF16:
        goto RF17;
STATE RF17:
        goto RF18;
STATE RF18:
        goto RF19;
STATE RF19:
        goto RF20;
STATE RF20:
        goto RF21;
STATE RF21:
        goto RF22;
STATE RF22:
        goto RF23;
STATE RF23:
        goto RF24;
STATE RF24:
        goto RF25;
STATE RF25:
        goto RF26;
STATE RF26:
        goto RF27;
STATE RF27:
        goto RF28;
STATE RF28:
        goto RF29;
STATE RF29:
        goto RF30;
STATE RF30:
        goto RF31;
STATE RF31:
        goto RF32;
STATE RF32:
        goto RF33;
STATE RF33:
        goto RF34;
STATE RF34:
        goto RF35;
STATE RF35:
        goto RF36;
STATE RF36:
        goto RF37;
STATE RF37:
        goto RF38;
```

Table A-1. i960® Jx Processor DRAM Controller ABEL File (Sheet 12 of 15)

```
STATE RF38:
        goto RF39;
STATE RF39:
        goto RF40;
STATE RF40:
        goto RF41;
STATE RF41:
        goto RF42;
STATE RF42:
        goto RF43;
STATE RF43:
        goto RF44;
STATE RF44:
        goto RF45;
STATE RF45:
        goto RF46;
STATE RF46:
        goto RF47;
STATE RF47:
        goto RF48;
STATE RF48:
        goto RF49;
STATE RF49:
        goto RF50;
STATE RF50:
        goto RF51;
STATE RF51:
        goto RF52;
STATE RF52:
        goto RF53;
STATE RF53:
        goto RF54;
STATE RF54:
        goto RF55;
STATE RF55:
        goto RF56;
STATE RF56:
        goto RF57;
STATE RF57:
        goto RF58;
STATE RF58:
        goto RF59;
STATE RF59:
        goto RF60;
STATE RF60:
        goto RF61;
STATE RF61:
        goto RF62;
```

```
STATE RF68:
        goto RF69;
STATE RF69:
        goto RF70;
STATE RF70:
        goto RF71;
STATE RF71:
        goto RF72;
STATE RF72:
        goto RF73;
STATE RF73:
        goto RF74;
STATE RF74:
        goto RF75;
STATE RF75:
        goto RF76;
STATE RF76:
        goto RF77;
STATE RF77:
        goto RF78;
STATE RF78:
        goto RF79;
STATE RF79:
        goto RF80;
STATE RF80:
        goto RF81;
STATE RF81:
        goto RF82;
STATE RF82:
        goto RF83;
STATE RF83:
        goto RF84;
STATE RF84:
        goto RF85;
STATE RF85:
        goto RF86;
STATE RF86:
        goto RF87;
```

Table A-1. i960® Jx Processor DRAM Controller ABEL File (Sheet 14 of 15)

```
STATE RF87:
        goto RF88;
STATE RF88:
        goto RF89;
STATE RF89
        :goto RF90;
STATE RF90:
        goto RF91;
STATE RF91:
        goto RF92;
STATE RF92:
        goto RF93;
STATE RF93:
        goto RF94;
STATE RF94:
        goto RF95;
STATE RF95:
        goto RF96;
STATE RF96:
        goto RF97;
STATE RF97:
        goto RF98;
STATE RF98:
        goto RF0 ;
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
" PRE_CAS used for the nCAS3:0 signals to conserve on P-Terms.
" PRE_CAS is asserted one 2X clock before actual CAS signal asserts. This was done
" because PRE_CAS is routed through another macrocell to produce nCAS
" with its associated nBE.
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
state_diagram nPRE_CAS   "nREF_CAS

        state SET_LOW:

               if((DRAM_STATES == REF4) & PH2 & DRAM_REF) then SET_HI " REF. DEASSERT
        else
               if(DRAM_STATES == DATA) then SET_HI " CAS DEASSERT
        else
               SET_LOW;


        state SET_HI:

                if((DRAM_STATES == IDLE) & PH1 & DRAM_REF) then SET_LOW " REF. ASSERT
        else
                if((DRAM_STATES == WAIT2) & !W_nR) #(DRAM_STATES==WAIT1)then SET_LOW
        else

                SET_HI;
```

Table A-1. i960® Jx Processor DRAM Controller ABEL File (Sheet 15 of 15)

```
"*****************************************************************************
EQUATIONS

!nREADY := (DRAM_STATES == WAIT1); " DRAM ready
DRAMACC = LATCHACC # !nADS & AD31 & !AD30 & AD29 & ! AD28; " Flags DRAM accesses

!nCAS0 := !nPRE_CAS & (!nBE0 # DRAM_REF);

!nCAS1 := !nPRE_CAS & (!nBE1 # DRAM_REF);

!nCAS2 := !nPRE_CAS & (!nBE2 # DRAM_REF);

!nCAS3 := !nPRE_CAS & (!nBE3 # DRAM_REF);

REFRESH_SYNC := (REFRESH == RF98);     " Used to synch refresh counter to 1X clock

"*****************************************************************************
"               INITIALIZE CONDITIONS AND CLOCKING
"*****************************************************************************

[LATCH_ACC,REFRESH_SYNC,nREADY,DRAMREF,DRAMBANK,BA3,BA2,Q3,Q2,Q1,Q0,nWE_OUT].clk = CLK1X;
[LATCH_ACC, REFRESH_SYNC, DRAMREF, DRAMBANK, BA3, BA2, Q3, Q2, Q1, Q0].re = !nRESET;
[nREADY, nWE_OUT].pr = !nRESET;

[nPRE_CAS, nCAS3, nCAS2, nCAS1, nCAS0, nRAS_1, nRAS_0, MUX_OUT].clk = CLK2X;
[nPRE_CAS, nCAS3, nCAS2, nCAS1, nCAS0, nRAS_1, nRAS_0].pr = !nRESET;
MUX_OUT.re = !nRESET;

[R6, R5, R4, R3, R2, R1, R0].clk = COM_CLK;  " REFRESH COUNTER
[R6, R5, R4, R3, R2, R1, R0].re = !nRESET;

end JXDRAM
```

Table A-2. Signal and Product Term Allocation

OUTPUT MACROCELLS		BURIED MACROCELLS	
Signal	**Product Terms**	**Signal**	**Product Terms**
nRAS_1	4	nPRECAS	8
nRAS_0	4	DRAMREF	2
nWE_OUT	2	LATCH_ACC	5
MUX_OUT	2	DRAM_ACC	2
BA3	3	DRAMBANK	6
BA2	3	REFRESH_SYNC	1
nREADY	1	Q3	3
nCAS_3	2	Q2	4
nCAS_2	2	Q1	5
nCAS_1	2	Q0	5
nCAS_0	2	R6	3
		R5	7
		R4	6
		R3	5
		R2	4
		R1	5
		R0	6

intel

AP-716

APPLICATION NOTE

80960Cx/80960Jx/80960Hx Architectural Comparison

January 29, 1996

Order Number: 272694-001

intel

AP-716

APPLICATION
NOTE

80960Cx/80960Jx/80960Hx
Architectural Comparison

January 29, 1996

Order Number: 272894-001

complete compatibility was not maintained. This document discusses in detail those differences between the 80960Cx, 80960Jx, and 80960Hx microprocessors. Suggestions are given for mitigating their impact on future designs.

This document also provides hints for developing applications which the 80960Cx processor can use now, and can be 80960Hx-ready in the future. The reader should note that the 80960Hx processor is **not drop-in compatible** with the for more detailed information.

1.1 i960® Processor Family

Table 1 summarizes the features of the 80960Cx, 80960Jx, and 80960Hx microprocessor families. Subsections which follow further describe each processor's features.

Table 1. Feature Summary of the 80960Cx, 80960Jx, and 80960Hx Processors (Sheet 1 of 2)

Feature	Processor		
	80960Cx	80960Jx	80960Hx
Core	Superscalar (maximum 3 inst/clk)	Scalar, clock doubled	Superscalar (max 3 inst/clk), clock doubled, clock tripled
External Bus	32-bit demultiplexed address and data	32-bit multiplexed address/data	32-bit demultiplexed address and data, parity on data
Instruction Cache	CA: 1 Kbyte, 2-way CF: 4 Kbytes, 2-way	4 Kbytes, 2-way	16 Kbytes, 4-way
Data Cache	CA: None CF: 1 Kbyte, direct map, write-through	2 Kbytes, direct map, write-through	8 Kbytes, 4-way, write-through
Data RAM	1 Kbyte, mapped from 000H to 3FFH	1 Kbyte, mapped from 0000H to 3FFH	2 Kbytes, mapped from 000H to 7FFH
Register Cache	5 frames, programmable to 15 frames (more than 5 uses Data RAM)	8 frames	5 frames, programmable to 15 frames (more than 5 uses Data RAM)
Memory-mapped Registers	No	Yes	Yes
Direct Memory Access (DMA) Controller	Yes	No	No

Table 1. Feature Summary of the 80960Cx, 80960Jx, and 80960Hx Processors (Sheet 2 of 2)

Feature	Processor		
	80960Cx	80960Jx	80960Hx
Interrupt Controller	Yes	Yes	Yes
Guarded Memory Unit	No	No	Yes
Timers	None	Two	Two
Power Supply	5V	5V or 3.3V	3.3V, 5V tolerant
JTAG	No	Yes	Yes

1.1.1 The 80960Cx Processor Family

The 80960Cx family of processors are superscalar implementations of the i960 microprocessor architecture, and feature demultiplexed, 32-bit address and data buses. The two members of this family, the CA and CF, can execute up to three instructions per clock cycle. Due to the high degree of parallelism and on-chip caches, these processors are well suited for high performance applications such as networking and high-end imaging.

The 80960CA includes a 1 Kbyte, two-way set associative instruction cache and 1 Kbyte of on-chip data RAM. Data RAM is located in the processor's address space from locations 0000.0000H to 0000.03FFH. The CF processor enhances the CA feature set by increasing the instruction cache size to 4 Kbytes, and adding a 1 Kbyte direct-mapped write-through data cache. As with the CA, the CF contains 1 Kbyte of on-chip data RAM located from 0000.0000H to 0000.03FFH.

The upper 16 Mbytes of the address space (FF00.0000H through FFFF.FFFFH) are reserved for implementation-specific functions. In general, an application must not access this space unless specifically required by the implementation. In the case of the CA and CF processors, the application must locate the Initialization Boot Record (IBR) at address FFFF.FF00H.

Integrated peripherals include the Interrupt Controller and Direct Memory Access (DMA) Controller. The DMA peripheral is not available on the 80960Jx and 80960Hx processors.

1.1.2 The 80960Jx Processor Family

The 80960Jx family of microprocessors represent the next generation of high-performance, low-cost processors based on the i960 architecture. 80960Jx processors feature a 32-bit multiplexed address and data bus; the core issues one instruction per clock cycle. The 80960Jx processor family supports 5V and 3.3V operation.

These processors provide a maximum 4 Kbyte, two-way set associative instruction cache, and a maximum 2 Kbyte, direct-mapped write-through data cache. The 80960Jx features 1 Kbyte of on-chip data RAM located from address 0000.0000H to 0000.03FFH.

The upper 16 Mbytes of the address space (FF00.0000H through FFFF.FFFFH) are reserved for implementation-specific functions. In general, an application must not access this space unless specifically required by the implementation. Memory-mapped control registers are located within this address space:

- user/supervisor accessible control registers are located from address FF00.0000H to FF00.7FFFH
- supervisor only accessible control registers from address FF00.8000H to FFFF.FFFFH

The application must locate the Initialization Boot Record (IBR) at address FEFF.FF30H.

Integrated peripherals include a 80960Cx-compatible Interrupt Controller, and two 80960Hx-compatible 32-bit timers. **The 80960Jx does not implement a DMA peripheral.**

1.1.3 The 80960Hx Processor Family

The 80960Hx family of microprocessors represent the next generation of very high-performance superscalar processors based on the i960 architecture. 80960Hx processors feature 32-bit demultiplexed address and data buses; the core may issue up to three instructions per clock cycle. The 80960Hx processor family is fabricated on a 3.3V process. All internal logic and memories operate at 3.3V; however, the pad-ring may be biased at 5V, meaning that the 80960Hx can operate in a 5V system. To do so, the system must provide a low-current 5V supply to bias the I/O buffers. In any case, the system must provide a 3.3V supply.

These processors provide a 16 Kbyte, four-way set associative instruction cache, and 8 Kbyte, four-way set-associative write-through data cache. The 80960Hx includes 2 Kbytes of on-chip data RAM located from address 0000.0000H to 0000.07FFH.

The upper 16 Mbytes of the address space (FF00.0000H through FFFF.FFFFH) are reserved for implementation-specific functions. In general, an application must not access this space unless specifically required by the implementation. Memory-mapped control registers are located within this address space:

- user/supervisor accessible control registers are located from address FF00.0000H to FF00.7FFFH
- supervisor only accessible control registers from address FF00.8000H to FFFF.FFFFH

The application must locate the Initialization Boot Record (IBR) at address FEFF.FF30H.

Integrated peripherals include a 80960Cx-compatible Interrupt Controller, two 80960Jx-compatible 32-bit timers, and the Guarded Memory Unit (GMU). **The 80960Hx does not implement a DMA peripheral.**

2.0 Initialization

This section outlines differences in initialization data structures and initialization procedures of the 80960Cx, 80960Jx, and 80960Hx. Only differences between these processor families are highlighted.

2.1 Initialization Data Structures

This section describes the differences between the necessary initialization data structures. Subsections include Section 2.1.1, Initialization Boot Record (IBR), Section 2.1.2, Processor Control Block (PRCB), Section 2.1.3, Fault Table, Section 2.1.4, Control Table, Section 2.1.5, Arithmetic Control Register Initial Image, and the rest.

2.1.1 Initialization Boot Record (IBR)

The 80960Cx, 80960Jx, and 80960Hx share similar initialization mechanisms. All processors fetch an Initialization Boot Record (IBR) from a fixed location in memory. The IBR contains the address of the first instruction to fetch and the pointer to the Processor Control Block (PRCB), which is the same format for all processors. The IBR structure is identical for the 80960Cx, 80960Jx, and 80960Hx, and is given in Figure 1. However, it is important to note that the interpretation of the **InitBusConx** bytes varies across implementations.

InitBusConx initializes:

- MCON0 on the 80960Cx
- PMCON14:15 on the 80960Jx
- PMCON15 on the 80960Hx

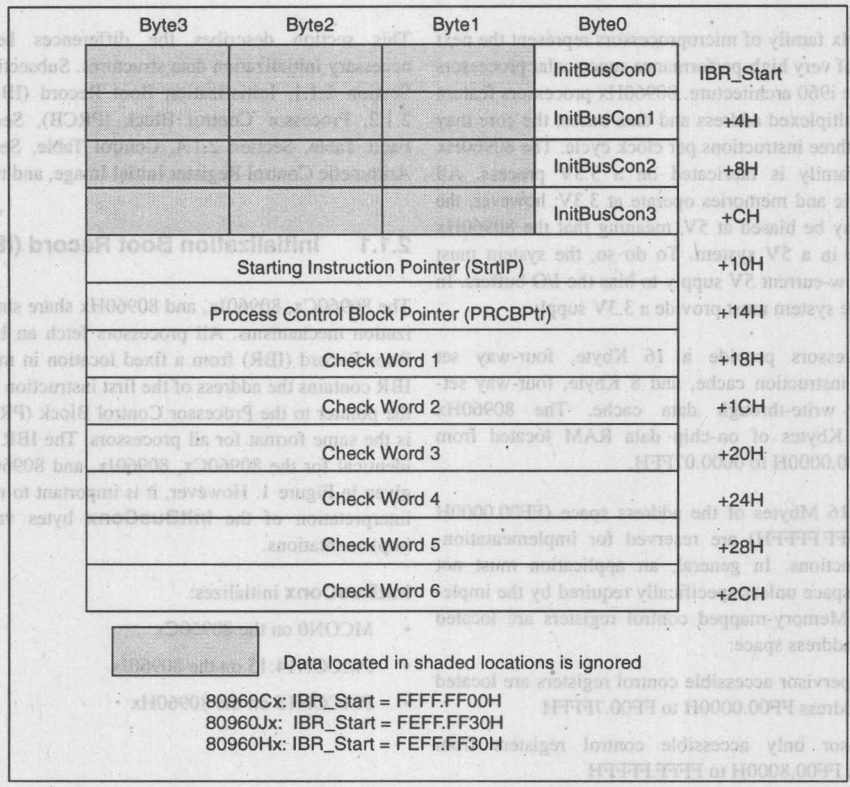


Figure 1. Initialization Boot Record (IBR) Structure

As indicated in Figure 1, the IBR contains the address of the first instruction to be fetched (**StrtIPx**), and the pointer to the PRCB (**PRCBPtrx**). The IBR also contains the initial bus configuration (**InitBusConx**), and check words for the bus confidence test. The algorithm used for the bus confidence test is identical for all processors.

The 80960Cx locates the IBR structure at address **FFFF.FF00H**; the 80960Jx and 80960Hx locate the IBR structure at address **FEFF.FF30H**. This was done so that a single boot ROM could hold two separate IBRs; one IBR for the 80960Cx, and one IBR for the 80960Jx or 80960Hx. This scheme works because of the way accesses to system memory are typically performed. Separate IBRs are necessary because of differences in initial register images, interpretation of the **InitBusConx** bytes, and control table structures. To clarify, refer to Figure 2.

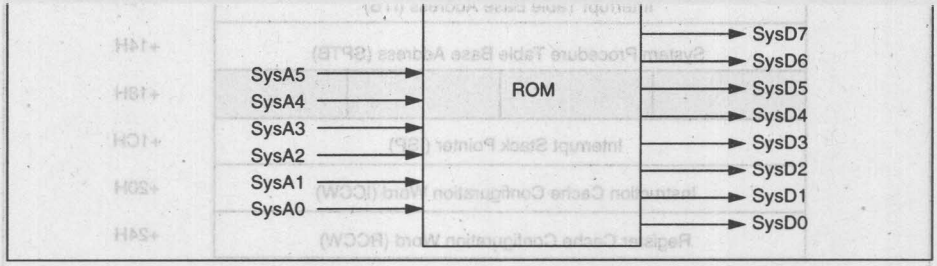


Figure 2. Simplified Typical Memory Decoding Logic

As can be seen from the figure above, system designers typically do not decode all high-order address pins when accessing memory. This allows for addresses to be aliased; logically different addresses will access the same physical memory.

Assume that the ROM in the figure above contains 64 Kbytes. In this case, when the 80960Cx accesses its IBR at FFFF.FF00H, address FF00H in the ROM is physically accessed. Likewise, when the 80960Jx or 80960Hx access the IBR at FEFF.FF30H, address FF30H in the ROM is physically accessed. Hence, both IBRs may be conveniently contained in the same ROM device. In the figure above, A24 can be ignored with the same results.

2.1.2 Processor Control Block (PRCB)

PRCB structure, illustrated in Figure 3, is identical for the 80960Cx, 80960Jx, and 80960Hx. Even though the PRCB structures are identical, the application designed to accommodate the 80960Cx and 80960Hx can be designed to provide separate control structures for each. This allows the application to define which control structures are common to the 80960Cx and 80960Hx, and which are unique.

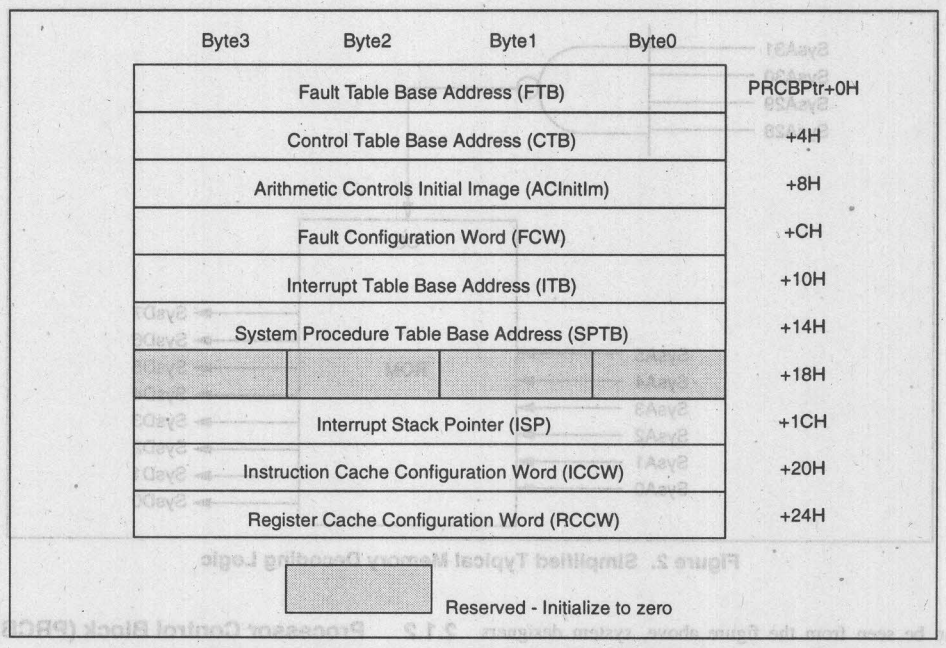


Figure 3. Processor Control Block (PRCB) Structure

As shown in Figure 1. Initialization Boot Record (IBR) Structure, the **PRCBPtr** is found in the IBR.

2.1.3 Fault Table

As indicated in Figure 3, the pointer to the fault table is found in the PRCB at offset 0H and is referred to as **FTB** (fault table base address). It may be initialized to any valid, non-reserved location in memory. The **FTB** points to the beginning of the fault table, which contains entries for the beginning address of each fault handler.

The fault tables for the 80960Cx, 80960Jx, and 80960Hx processors are identical, with the exception that the 80960Hx adds an entry for machine faults and an additional fault sub-type to support the Guarded Memory Unit. Machine faults are generated in response to parity errors. This is illustrated in Figure 4.

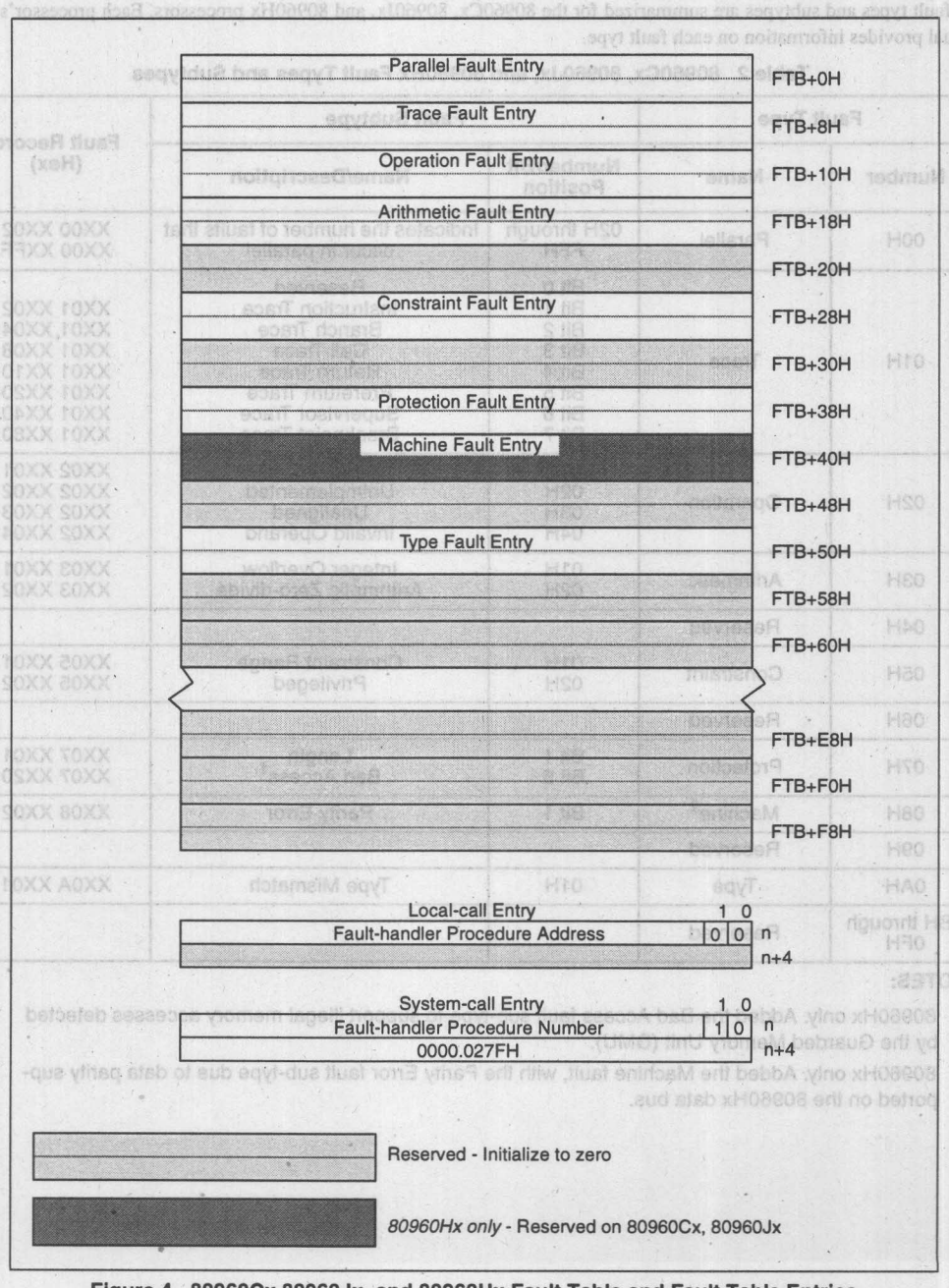


Figure 4. 80960Cx 80960Jx, and 80960Hx Fault Table and Fault Table Entries

The fault types and subtypes are summarized for the 80960Cx, 80960Jx, and 80960Hx processors. Each processor's users manual provides information on each fault type.

Table 2. 80960Cx, 80960Jx, and 80960Hx Fault Types and Subtypes

Fault Type		Fault Subtype		Fault Record (Hex)
Number	Name	Number/Bit Position	Name/Description	
00H	Parallel	02H through FFH	Indicates the number of faults that occur in parallel	XX00 XX02 XX00 XXFF
01H	Trace	Bit 0 Bit 1 Bit 2 Bit 3 Bit 4 Bit 5 Bit 6 Bit 7	Reserved Instruction Trace Branch Trace Call Trace Return Trace Prereturn Trace Supervisor Trace Breakpoint Trace	 XX01 XX02 XX01 XX04 XX01 XX08 XX01 XX10 XX01 XX20 XX01 XX40 XX01 XX80
02H	Operation	01H 02H 03H 04H	Invalid Opcode Unimplemented Unaligned Invalid Operand	XX02 XX01 XX02 XX02 XX02 XX03 XX02 XX04
03H	Arithmetic	01H 02H	Integer Overflow Arithmetic Zero-divide	XX03 XX01 XX03 XX02
04H	Reserved			
05H	Constraint	01H 02H	Constraint Range Privileged	XX05 XX01 XX05 XX02
06H	Reserved			
07H	Protection	Bit 1 Bit 5	Length Bad Access[1]	XX07 XX01 XX07 XX20
08H	Machine[2]	Bit 1	Parity Error	XX08 XX02
09H	Reserved			
0AH	Type	01H	Type Mismatch	XX0A XX01
0BH through 0FH	Reserved			

NOTES:

1. 80960Hx only. Added the Bad Access fault sub-type to support illegal memory accesses detected by the Guarded Memory Unit (GMU).
2. 80960Hx only. Added the Machine fault, with the Parity Error fault sub-type due to data parity supported on the 80960Hx data bus.

2.1.4 Control Table

As shown in Figure 3, the pointer to the control table is found in the PRCB at offset 4H and is referred to as **CTB** (control table base address). It may be initialized to any valid, non-reserved location in memory. **CTB** points to the beginning of the control table, which contains entries for the initial values of the memory configuration, breakpoint, interrupt control, and additional registers.

The control table for the 80960Cx, 80960Jx, and 80960Hx processors is shown in Table 3 Unique control tables are required for each processor, due to differences in the region and bus control registers. Refer to each processor's users manual for additional details

Table 3. Control Table for the 80960Cx, 80960Jx, and 80960Hx Processors (Sheet 1 of 3)

Offset	Description	Group
CTB + 00H	80960Cx: Instruction Breakpoint 0 (IPB0) 80960Jx: Reserved 80960Hx: Reserved	0H
CTB + 04H	80960Cx: Instruction Breakpoint 1 (IPB1) 80960Jx: Reserved 80960Hx: Reserved	
CTB + 08H	80960Cx: Data Address Breakpoint 0 (DAB0) 80960Jx: Reserved 80960Hx: Reserved	
CTB + 0CH	80960Cx: Data Address Breakpoint 1 (DAB1) 80960Jx: Reserved 80960Hx: Reserved	
CTB + 10H	80960Cx: Interrupt Map 0 (IMAP0) 80960Jx: Same 80960Hx: Same	1H
CTB + 14H	80960Cx: Interrupt Map 1 (IMAP1) 80960Jx: Same 80960Hx: Same	
CTB + 18H	80960Cx: Interrupt Map 2 (IMAP2) 80960Jx: Same 80960Hx: Same	
CTB + 1CH	80960Cx: Interrupt Control (ICON) 80960Jx: Same 80960Hx: Same	
CTB + 20H	80960Cx: Memory Region 0 Configuration (MCON0) 80960Jx: Physical Memory Region 0:1 Configuration (PMCON0:1) 80960Hx: Physical Memory Region 0 Configuration (PMCON0)	2H
CTB + 24H	80960Cx: Memory Region 1 Configuration (MCON1) 80960Jx: Reserved (initialize to zero) 80960Hx: Physical Memory Region 1 Configuration (PMCON1)	
CTB + 28H	80960Cx: Memory Region 2 Configuration (MCON2) 80960Jx: Physical Memory Region 2:3 Configuration (PMCON2:3) 80960Hx: Physical Memory Region 2 Configuration (PMCON2)	
CTB + 2CH	80960Cx: Memory Region 3 Configuration (MCON3) 80960Jx: Reserved (initialize to zero) 80960Hx: Physical Memory Region 3 Configuration (PMCON3)	

Table 3. Control Table for the 80960Cx, 80960Jx, and 80960Hx Processors (Sheet 2 of 3)

Offset	Description	Group
CTB + 30H	80960Cx: Memory Region 4 Configuration (MCON4) 80960Jx: Physical Memory Region 4:5 Configuration (PMCON4:5) 80960Hx: Physical Memory Region 4 Configuration (PMCON4)	3H
CTB + 34H	80960Cx: Memory Region 5 Configuration (MCON5) 80960Jx: Reserved (initialize to zero) 80960Hx: Physical Memory Region 5 Configuration (PMCON5)	
CTB + 38H	80960Cx: Memory Region 6 Configuration (MCON6) 80960Jx: Physical Memory Region 6:7 Configuration (PMCON6:7) 80960Hx: Physical Memory Region 6 Configuration (PMCON6)	
CTB + 3CH	80960Cx: Memory Region 7 Configuration (MCON7) 80960Jx: Reserved (initialize to zero) 80960Hx: Physical Memory Region 7 Configuration (PMCON7)	
CTB + 40H	80960Cx: Memory Region 8 Configuration (MCON8) 80960Jx: Physical Memory Region 8:9 Configuration (PMCON8:9) 80960Hx: Physical Memory Region 8 Configuration (PMCON8)	4H
CTB + 44H	80960Cx: Memory Region 9 Configuration (MCON9) 80960Jx: Reserved (initialize to zero) 80960Hx: Physical Memory Region 9 Configuration (PMCON9)	
CTB + 48H	80960Cx: Memory Region 10 Configuration (MCON10) 80960Jx: Physical Memory Region 10:11 Configuration (PMCON10:11) 80960Hx: Physical Memory Region 10 Configuration (PMCON10)	
CTB + 4CH	80960Cx: Memory Region 11 Configuration (MCON11) 80960Jx: Reserved (initialize to zero) 80960Hx: Physical Memory Region 11 Configuration (PMCON11)	
CTB + 50H	80960Cx: Memory Region 12 Configuration (MCON12) 80960Jx: Physical Memory Region 12:13 Configuration (PMCON12:13) 80960Hx: Physical Memory Region 12 Configuration (PMCON12)	5H
CTB + 54H	80960Cx: Memory Region 13 Configuration (MCON13) 80960Jx: Reserved (initialize to zero) 80960Hx: Physical Memory Region 13 Configuration (PMCON13)	
CTB + 58H	80960Cx: Memory Region 14 Configuration (MCON14) 80960Jx: Physical Memory Region 14:15 Configuration (PMCON14:15) 80960Hx: Physical Memory Region 14 Configuration (PMCON14)	
CTB + 5CH	80960Cx: Memory Region 15 Configuration (MCON15) 80960Jx: Reserved (initialize to zero) 80960Hx: Physical Memory Region 15 Configuration (PMCON15)	

Table 3. Control Table for the 80960Cx, 80960Jx, and 80960Hx Processors (Sheet 3 of 3)

Offset	Description	Group
CTB + 60H	80960Cx: Reserved (initialize to zero) 80960Jx: Reserved (initialize to zero) 80960Hx: Reserved (initialize to zero)	6H
CTB + 64H	80960Cx: Breakpoint Control (BPCON) 80960Jx: Same 80960Hx: Same	
CTB + 68H	80960Cx: Trace Controls (TC) 80960Jx: Same 80960Hx: Same	
CTB + 6CH	80960Cx: Bus Configuration Control (BCON) 80960Jx: Same 80960Hx: Same	

NOTES:

1. CTB = Control Table Base address, found in the Processor Control Block (PRCB).
2. Differences exist between the processors in the format of the IMAPx and ICON registers.
3. Differences exist between the processors in the format of the BCON, MCON, and PMCON registers. Note that MCON0, PMCON14:15, and PMCON15 are initialized from the IBR at reset.

2.1.5 Arithmetic Control Register Initial Image

The arithmetic control register (AC) is defined identically for the 80960Cx, 80960Jx, and 80960Hx processors. The AC initial image is programmed in the PRCB at offset 8H, and is shown as **ACinitIm** in Figure 3, Processor Control Block (PRCB) Structure (pg. 6).

2.1.6 Fault Configuration Word

The fault configuration word is defined identically for the 80960Cx, 80960Jx, and 80960Hx processors. The fault configuration word is programmed in the PRCB at offset CH, and is shown as **FCW** in Figure 3, Processor Control Block (PRCB) Structure (pg. 6).

2.1.7 Interrupt Table

Recall that the pointer to the interrupt table base address is found in the PRCB at offset 10H. The interrupt table base address pointer is identified as **ITB** (interrupt table base address), and may be initialized to any valid, non-reserved location in memory. **ITB** points to the beginning of the interrupt table, which contains entries for the starting addresses (interrupt vectors) of all interrupt service routines.

The formats of the interrupt tables for the 80960Cx, 80960Jx, and 80960Hx processors are identical. Interrupt vectors may be internally cached in on-chip data RAM in the same manner as well. The system may provide unique versions of the interrupt table for the 80960Cx and 80960Hx processors, so that the interrupt enable (**inten**) and disable (**intdis**) instructions may be used, which are not available on the 80960Cx processor.

2.1.8 System Procedure Table

As shown in Figure 3, Processor Control Block (PRCB) Structure (pg. 6), the pointer to the system procedure table is found in the PRCB at offset 14H. It is identified as **SPTB** (system procedure table base address), and may be initialized to any valid, non-reserved location in memory. **SPTB** points to the beginning of the system procedure table, which contains entries for the starting addresses system procedures. System procedures may be invoked by executing the call system (**calls**) instruction, or by the fault generator.

The system procedure table formats for all three processors are identical.

The instruction cache configuration word (ICCW) is defined identically for all three processors. The ICCW initial image is programmed in the PRCB at offset 20H, and is shown as **ICCW** in Figure 3.

2.1.11 Register Cache Configuration Word

The register cache configuration word (RCCW) initial image is programmed in the PRCB at offset 24H, and is shown as **RCCW** in Figure 3, Processor Control Block (PRCB) Structure (pg. 6). The 80960Jx and 80960Hx processors add additional programmability in addition to

2.2 Register Values after Reset/Reinitialization

This section describes the values of the architectural state of the 80960Cx, 80960Jx, and 80960Hx processors after hardware reset and software reinitialization.

2.2.1 Common Reset/Reinitialized States

Table 4 shows the inital state values held in common by all three processors after hardware reset and software reinitialization.

Table 4. Common State of the i960® Processors after Reset/Reinitialization (Sheet 1 of 2)

Register	Value after Hardware Reset	Value after Software Reinitialization
g0	Device ID	Device ID
AC	AC initial image at **PRCBptr + 8H**	AC initial image at **PRCBPtr + 8H**
TC	TC initial image at **CTB + 68H**	TC initial image at **CTB + 68H**
FP (g15)	**ISP (PRCBPtr + 1CH)**	**ISP (PRCBPtr + 1CH)**
PFP (r0)	Undefined	Value before Software Reinitialization
SP (r1)	**ISP** + 40H ([**PRCBPtr + 1CH**] + 40H)	**ISP** + 40H ([**PRCBPtr + 1CH**] + 40H)
RIP (r2)	Undefined	Undefined
IPND	Undefined	Value before Software Reinitialization
IMSK	00H	00H
IMAP0	CTB + 10H	CTB + 10H
IMAP1	CTB + 14H	CTB + 14H
IMAP2	CTB + 18H	CTB + 18H
ICON	CTB + 1CH	CTB + 1CH
80960Cx: MCON0 80960Jx: PMCON0:1 80960Hx: PMCON0	CTB + 20H	CTB + 20H
80960Cx: MCON1 80960Hx: PMCON1	CTB + 24H	CTB + 24H
80960Cx: MCON2 80960Jx: PMCON2:3 80960Hx: PMCON2	CTB + 28H	CTB + 28H
80960Cx: MCON3 80960Hx: PMCON3	CTB + 2CH	CTB + 2CH
80960Cx: MCON4 80960Jx: PMCON4:5 80960Hx: PMCON4	CTB + 30H	CTB + 30H
80960Cx: MCON5 80960Hx: PMCON5	CTB + 34H	CTB + 34H
80960Cx: MCON6 80960Jx: PMCON6:7 80960Hx: PMCON6	CTB + 38H	CTB + 38H
80960Cx: MCON7 80960Hx: PMCON7	CTB + 3CH	CTB + 3CH
80960Cx: MCON8 80960Jx: PMCON8:9 80960Hx: PMCON8	CTB + 40H	CTB + 40H
80960Cx: MCON9 80960Hx: PMCON9	CTB + 44H	CTB + 44H
80960Cx: MCON10 80960Jx: PMCON10:11 80960Hx: PMCON10	CTB + 48H	CTB + 48H

Table 4. Common State of the i960® Processors after Reset/Reinitialization (Sheet 2 of 2)

Register	Value after Hardware Reset	Value after Software Reinitialization
80960Cx: MCON11 80960Hx: PMCON11	CTB + 4CH	CTB + 4CH
80960Cx: MCON12 80960Jx: PMCON12:13 80960Hx: PMCON12	CTB + 50H	CTB + 50H
80960Cx: MCON13 80960Hx: PMCON13	CTB + 54H	CTB + 54H
80960Cx: MCON14 80960Jx: PMCON14:15 80960Hx: PMCON14	CTB + 58H	CTB + 58H
80960Cx: MCON15 80960Hx: PMCON15	CTB + 5CH	CTB + 5CH
BCON	CTB + 6CH	CTB + 6CH

2.2.2 Differences in the Reset/Reinitialized State

Table 5 shows the differences in the state of the 80960Cx, 80960Jx, and 80960Hx processors after hardware reset and software reinitialization.

Table 5. Differences in State of the i960® Processors after Reset/Reinitialization (Sheet 1 of 2)

Register	Value after Hardware Reset	Value after Software Reinitialization	Processor		
			Cx	Jx	Hx
PC	C01F.2002H	C01F.2002H	X		
	001F.2002H	001F.2002H		X	X
DMAC (sf2)	00H	00H	X		
CCON (sf2)	TBD [1]	TBD [1]			X
BPCON	**CTB + 64H**	**CTB + 64H**	X		
	0000.0000H	0000.0000H		X	X
LMARx	Undefined	Value before Software Reinitialization		X	X
LMMRx	Bit 0 = 0; bits 1 - 31 = undefined	Bit 0 = 0; bits 1 - 31 = undefined [3]		X	X
DLMCON	Bit 0 = bit 7 of byte **InitBusCon**$_3$; bit 1 = 0; bits 2 - 31 = undefined	Bit 0 = value before warm reset; bit 1 = 0; bits 2 - 31 = undefined		X	X
TRRx	Undefined	Value before software reinitialization		X	X
TCRx	Undefined	Value before software reinitialization		X	X
TMRx	Bits 1 - 6 = 0; bits 0, 7 - 31 = undefined [2]	Bits 1 - 6 = 0; bits 0, 7 - 31 = undefined [2]		X	X

MDUB0 thru MDUB5	Undefined	Value before Software Reinitialization			X
MDLB0 thru MDLB5	Undefined	Value before Software Reinitialization			X
IPB0	CTB + 00H	CTB + 00H	X		
	0000.0000H	0000.0000H		X	X
IPB1	CTB + 04H	CTB + 04H	X		
	0000.0000H	0000.0000H		X	X
DAB0	CTB + 08H	CTB + 08H	X		
	0000.0000H	0000.0000H		X	X
DAB1	CTB + 0CH	CTB + 0CH	X		
	0000.0000	0000.0000H		X	X
IPB2 thru IPB5	0000.0000H	Value before Software Reinitialization			X
DAB2 thru DAB5	0000.0000H	Value before Software Reinitialization			X
XBPCON	0000.0000H	Value before Software Reinitialization			X
DEVICEID	Device Identification	Device Identification		X	X

NOTES:

1. Bit 31 = 1, all others = 0
2. Bits 1-5 = 0, bits 0, 6-31 = undefined
3. LMMRx retains value (Jx only) otherwise, bit 0 = 0, bits 1-31 = undefined

2.3 Initial Bus Configuration

As indicated in Figure 1, Initialization Boot Record (IBR) Structure, The **InitBusConx** bytes are used for the initial configuration of the bus controller. These bytes configure the bus controller during the interim period from the beginning of initialization until the remainder of the configuration data structures are loaded from external memory to fully configure the bus controller. Each processor uses these bytes to initialize the appropriate memory control register. Since each processor has unique memory control register formats, these bytes are interpreted differently for each processor. For more information on memory control, refer to Section 2.3.4, External Bus Configuration.

2.3.1 80960Cx Processor

The first four words of the IBR are fetched from memory with the most relaxed bus parameters, specifically:

- Non-burst
- Non-pipelined
- Ready disabled
- 8-bit bus width
- Little endian byte order
- $N_{RAD} = 31$
- $N_{RDD} = 3$
- $N_{WAD} = 31$
- $N_{WDD} = 3$
- $N_{XDA} = 3$

The 80960Cx provides memory control with 16 Memory Control registers, named MCON15:0. The MCON registers define the physical attributes of the external memory system: wait state profiles, bus width, pipelining, bursting, $\overline{\text{READY}}$ enable, etc. MCON0 controls external accesses to addresses **0000.0000H** through **0FFF.FFFFH**, MCON1 controls accesses **1000.0000H** through **1FFF.FFFFH**, and so forth. In this manner, the MCON registers control 256 Mbyte regions of memory. The bus controller interface is discussed in further detail in Section 2.3.4.1, External Memory Configuration on the 80960Cx Processor.

As the first four words of the IBR are read from external memory, the low-order byte from each word is loaded into MCON0. The exact order in which these words are read is implementation dependent. This means that **InitBusCon0** is loaded into $MCON0_0$, **InitBusCon1** into $MCON0_1$, and so forth. The most significant byte of all MCONx registers is reserved, and must be programmed to zero (**InitBusCon3** must be zero). Once MCON0 is configured, all subsequent external bus accesses are controlled by this register until the remaining MCONx and Bus Control (BCON) registers are configured. After the MCONx registers are configured, they are marked valid and then control accesses to their respective memory regions.

Typically, the IBR is located in ROM memory; MCON0 is initially loaded with the appropriate parameters to access this memory.

2.3.2 80960Jx Processor

The first four words of the IBR are fetched from memory with the most relaxed bus parameters, specifically:

- 8-bit bus width

Since there is no internal wait state generator on the 80960Jx, the external memory system must return $\overline{\text{READY}}$ as appropriate.

The 80960Jx allows enhanced memory control by providing eight Physical Memory Control registers (PMCONx:x+1), two Logical Memory Control register pairs (LMCONx), and the Default Logical Memory Control register (DLMCON). The PMCONx:x+1 registers define the bus width of the external memory system. The LMCONx register pairs define the logical characteristics of the external memory system which, in the case of the 80960Jx, is simply cacheability. The DLMCON register defines the default logical characteristics of the external memory system for addresses not specifically covered by the LMCONx register pairs. This means that the DLMCON register controls the default cacheability characteristic of external memory, as well as the globally controlling byte order (big- or little-endian). Byte order on the 80960Jx cannot be controlled by LMCONx register pairs on a region-by-region basis; the memory space is uniformly big- or little-endian.

PMCON0:1 controls the bus width for external accesses to addresses **0000.0000H** through **1FFF.FFFFH**, PMCON2:3 controls accesses **2000.0000H** through **3FFF.FFFFH**, and so forth. In this manner, the PMCONx:x+1 registers control 512 Mbyte regions of memory. The LMCON registers may be programmed to control a variable range of addresses. The bus controller interface is discussed in further detail in Section 2.3.4.2, External Memory Configuration on the 80960Jx Processor.

As the first four words of the IBR are read from external memory, the low-order byte from each word is loaded into memory configuration register 14:15 (PMCON14:15). The exact order in which these words are read is implementation dependent. This means that **InitBusCon0** is loaded into PMCON14:15_0, **InitBusCon1** into PMCON14:15_1, and so forth. Once PMCON14:15 is configured, all subsequent external bus accesses are controlled by this register until the remaining PMCONx:x+1 and Bus Control (BCON) registers are initialized. After the PMCONx:x+1 registers are initialized, they are marked valid and then control accesses to their respective memory regions. Bit 7 of **InitBusCon3** is loaded into bit 0 of the DLMCON register; this bit controls data endianess for the entire 32-bit memory space.

Typically, the IBR is located in ROM memory, and PMCON14:15 is initially loaded with the appropriate parameters to access this memory. Note that PMCON14:15 is initialized first from the IBR, as opposed to PMCON0:1 (or as MCON0 is on the 80960Cx). This makes sense since the IBR is fixed in memory region 15; accesses to this memory region are normally controlled by PMCON14:15, not PMCON0:1. On the 80960Cx, system programmers were often required to reconfigure MCON0 (which controls memory region zero) if the memory profile for this region did not match that of memory region 15.

2.3.3 80960Hx Processor

The first four words of the IBR are fetched from memory with the most relaxed bus parameters:

- Non-burst
- Non-pipelined
- Ready disabled
- 8-bit bus width
- Little endian byte order
- $N_{RAD} = 31$
- $N_{RDD} = 3$
- $N_{WAD} = 31$
- $N_{WDD} = 3$
- $N_{XDA} = 15$

The possible number of N_{XDA} wait states is increased from three on the 80960Cx, to fifteen on the 80960Hx.

The 80960Hx allows enhanced memory control by providing:

- 16 Physical Memory Control registers (PMCONx)
- 15 Logical Memory Control register pairs (LMCONx)
- Default Logical Memory Control register (DLMCON)

The PMCONx registers define the physical attributes of the external memory system: wait state profiles, parity, bus width, pipelining, bursting, $\overline{\text{READY}}$ enable, etc. The LMCONx register pairs define the logical characteristics of the external memory system: data endianess, data cache independently invalidatable regions, and cacheability. The DLMCON register defines the default logical characteristics of the memory system for addresses not specifically covered by the LMCONx register pairs.

PMCON0 controls external accesses to addresses **0000.0000H** through **0FFF.FFFFH**, PMCON1 controls accesses **1000.0000H** through **1FFF.FFFFH**, and so forth. In this manner, the PMCONx registers control 256 Mbyte regions of memory. The LMCONx register pairs may be programmed to control a variable range of addresses. PMCON registers (80960Hx) and MCON registers (80960Cx) are roughly analogous in functionality, if the endianess and cacheability bits are ignored. The bus controller interface is discussed in further detail in Section 2.3.4.3, External Memory Configuration on the 80960Hx Processor.

As the first four words of the IBR are successively read from external memory, the low-order byte from each word of the IBR is successively loaded into each byte of Memory Configuration register 15 (PMCON15). This means that **InitBusCon0** gets loaded into PMCON15_0, **InitBusCon1** into PMCON15_1, and so forth. Once PMCON15 is configured, all subsequent external bus accesses are controlled by this register until the remaining PMCONx and Bus Control (BCON) registers are initialized. After all PMCONx registers are initialized, they are marked valid and then control accesses to their respective memory regions.

Typically, the IBR is located in ROM memory, and PMCON15 is initially loaded with the appropriate parameters to access this memory. Note that PMCON15 is initialized first from the IBR, as opposed to PMCON0 (or as MCON0 is on the 80960Cx). This makes sense since the IBR is fixed in memory region 15; accesses to this memory region are normally controlled by PMCON15, not PMCON0. On the 80960Cx, system programmers were often required to reconfigure MCON0 (which controls

memory region zero) if the memory profile for this region did not match that of memory region 15.

2.3.4 External Bus Configuration

This section describes differences in the external bus configuration of the 80960Cx, 80960Jx, and 80960Hx processors. Specifically, this section outlines the differences in programming the Memory Region Configuration (MCON), Physical Memory Region Configuration (PMCON), Logical Memory Template Address (LMAR), Logical Memory Template Mask (LMMR), Default Logical Memory Configuration (DLMCON), and Bus Control (BCON) registers.

2.3.4.1 External Memory Configuration on the 80960Cx Processor

The 80960Cx processor's MCON registers define the physical nature of the memory system (wait state profile, ready enable, burst, pipeline, bus width), and logical nature (byte order, cacheability - CF only). There are 16 MCON registers, one for each 256-Mbyte physical memory region. These registers are labelled MCON0 through MCON15. The format of the MCON register is shown in Figure 5.

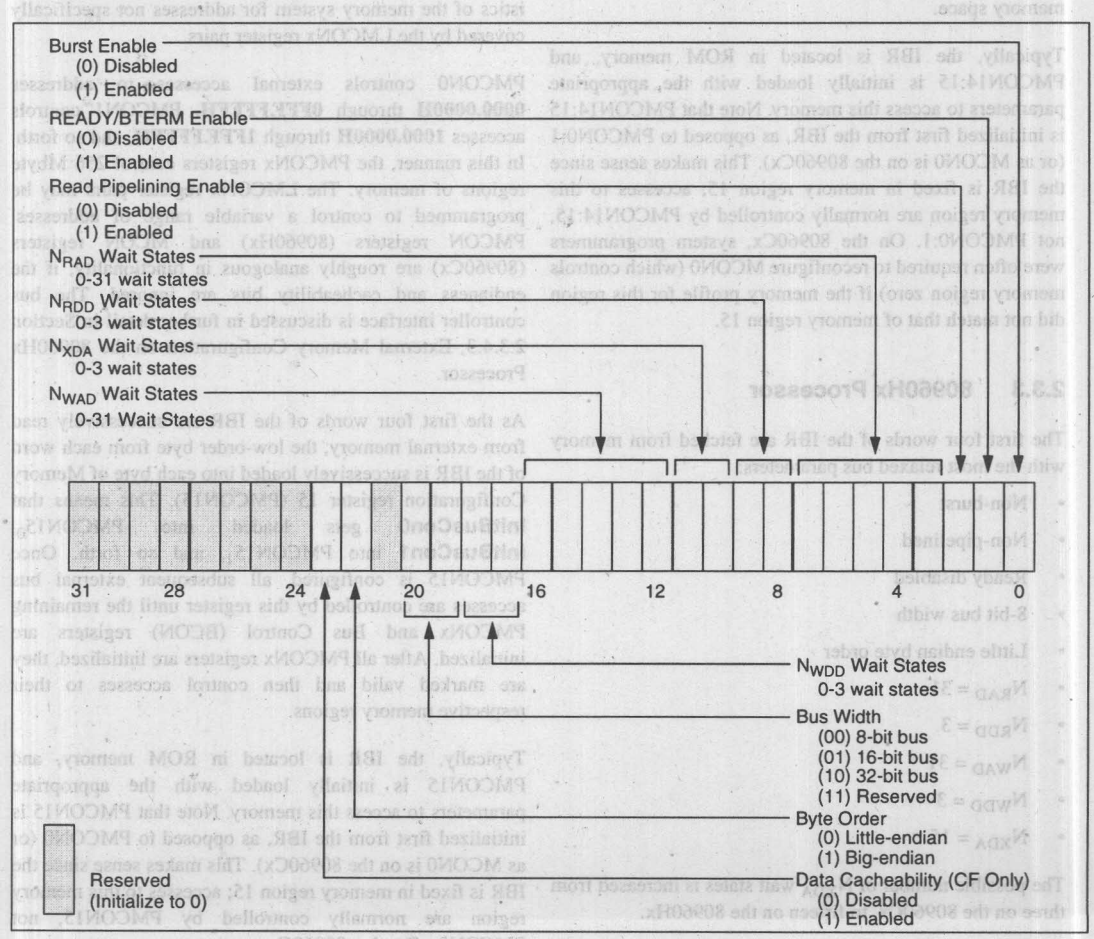


Figure 5. 80960Cx MCON Register Format

As shown in Figure 6, The Bus Control Register (BCON) mainly serves to validate the values stored in the MCON registers (region table). Immediately after a hardware reset, the region table is marked invalid in the BCON register. Whenever the region table is marked invalid in the BCON register (bit 0 = bit 0), the Bus Control Unit (BCU) uses the physical memory parameters found in MCON0 for accesses to all regions (Recall that MCON0 is initialized from data found in the IBR). Once the entire region table is loaded by reset microcode, the region table is marked valid in the BCON register (bit 0 = 1), and the BCU uses the appropriate user programmed values.

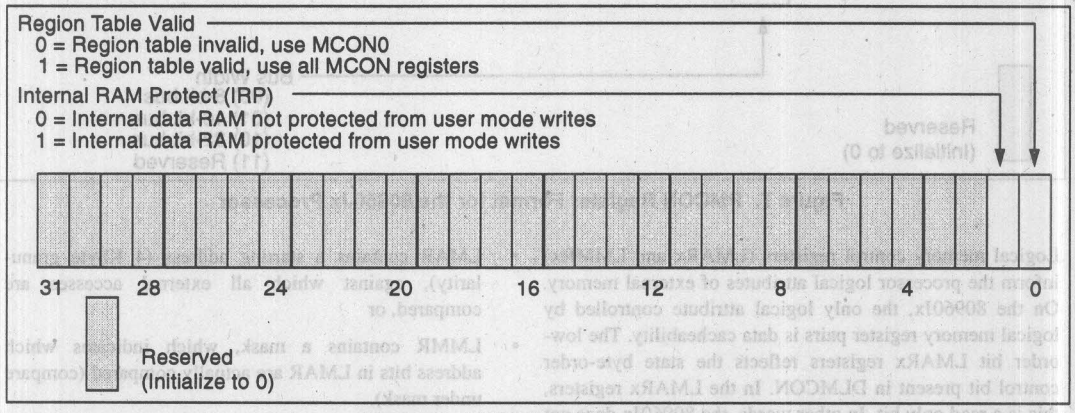


Figure 6. 80960Cx BCON Register

2.3.4.2 External Memory Configuration on the 80960Jx Processor

Control of physical and logical memory characteristics is divided among several registers on the 80960Jx:

- Physical Memory Region Control (PMCON)
- Logical Memory Template Address (LMAR)
- Logical Memory Template Mask (LMMR)
- Default Logical Memory Configuration (DLMCON) registers

There are eight PMCON registers, one for each 512 Mbyte physical memory region on the 80960Jx (PMCON0:1 through PMCON14:15). There are two pairs of LMAR/LMMR registers (LMAR0, LMAR1, LMMR0, LMMR1), and one DLMCON register.

PMCON registers control the bus width on the 80960Jx processor. This is shown in Figure 7.

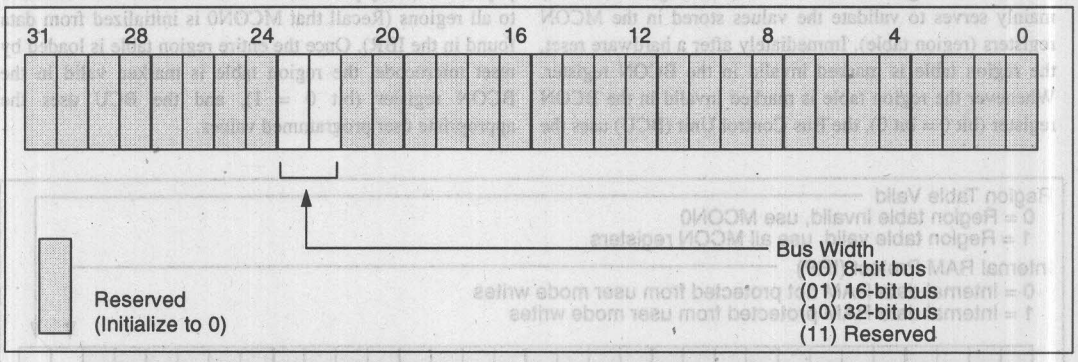


Figure 7. PMCON Register Format for the 80960Jx Processor

Logical memory control registers (LMARx and LMMRx) inform the processor logical attributes of external memory. On the 80960Jx, the only logical attribute controlled by logical memory register pairs is data cacheability. The low-order bit LMARx registers reflects the state byte-order control bit present in DLMCON. In the LMARx registers, this is a read only bit. In other words, the 80960Jx does not support multiple byte-order profiles by logical region; it implements only a homogeneous model. The entire memory map is either big-endian or little-endian.

Logical memory regions are created using the LMAR/LMMR register pair. Either:

- LMAR contains a starting address (4 Kbyte granularity), against which all external accesses are compared, or
- LMMR contains a mask, which indicates which address bits in LMAR are actually compared (compare under mask).

In this manner, one register pair can be used to create multiple logical regions, aliased on an arbitrary power-of-two boundary, by making some combination of upper memory bits "don't care." The format of the LMAR and LMMR registers is shown in Figure 8.

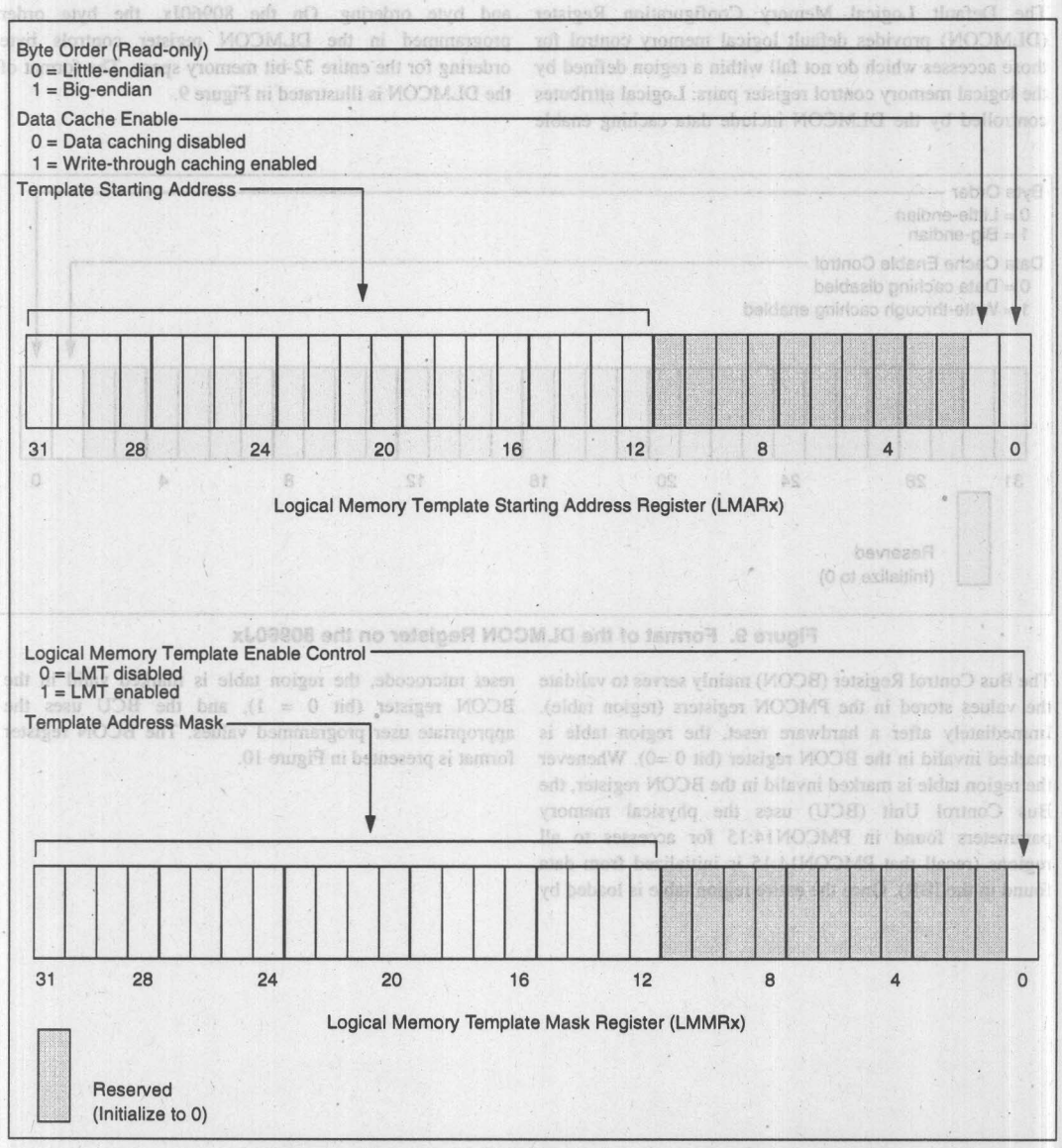


Figure 8. Format of LMARx and LMMRx Pairs on the 80960Jx Processor

The Default Logical Memory Configuration Register (DLMCON) provides default logical memory control for those accesses which do not fall within a region defined by the logical memory control register pairs. Logical attributes controlled by the DLMCON include data caching enable and byte ordering. On the 80960Jx, the byte order programmed in the DLMCON register controls byte ordering for the entire 32-bit memory space. The format of the DLMCON is illustrated in Figure 9.

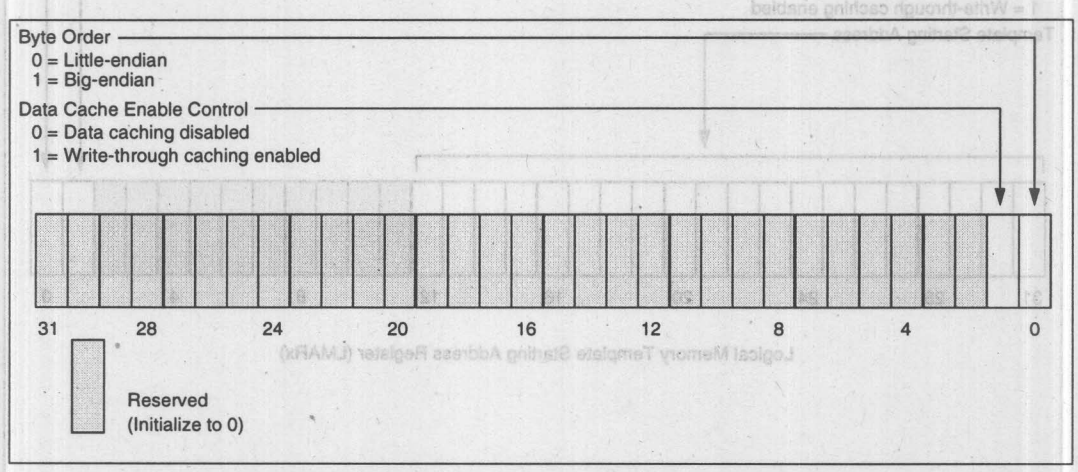


Figure 9. Format of the DLMCON Register on the 80960Jx

The Bus Control Register (BCON) mainly serves to validate the values stored in the PMCON registers (region table). Immediately after a hardware reset, the region table is marked invalid in the BCON register (bit 0 =0). Whenever the region table is marked invalid in the BCON register, the Bus Control Unit (BCU) uses the physical memory parameters found in PMCON14:15 for accesses to all regions (recall that PMCON14:15 is initialized from data found in the IBR). Once the entire region table is loaded by reset microcode, the region table is marked valid in the BCON register (bit 0 = 1), and the BCU uses the appropriate user programmed values. The BCON register format is presented in Figure 10.

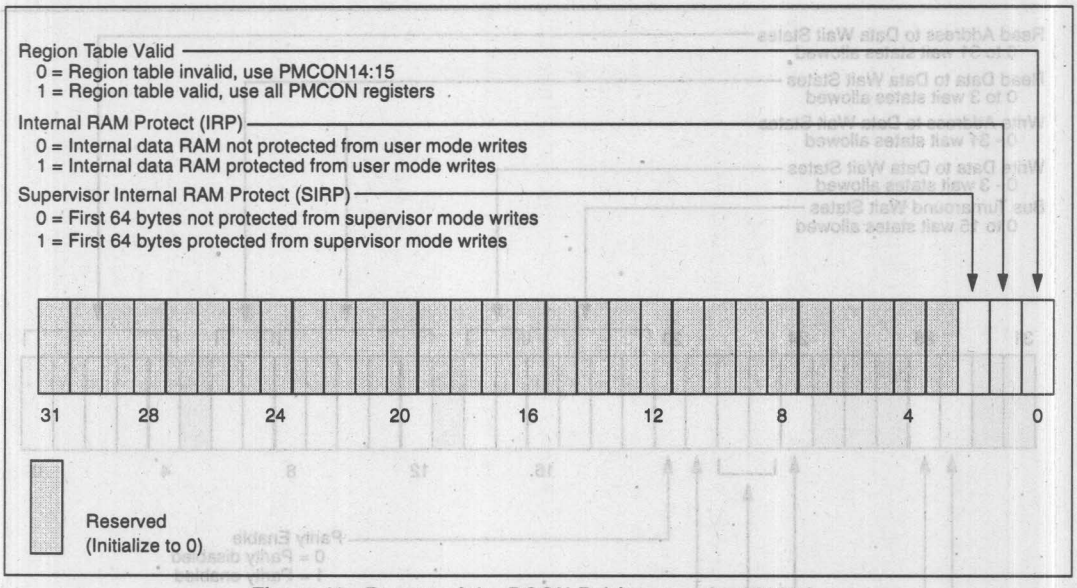


Figure 10. Format of the BCON Register on the 80960Jx

2.3.4.3 External Memory Configuration on the 80960Hx Processor

Control of physical and logical memory characteristics is divided among several registers on the 80960Hx:

- Physical Memory Region Control (PMCON)
- Logical Memory Template Address (LMAR)
- Logical Memory Template Mask (LMMR)
- Default Logical Memory Configuration (DLMCON) registers

There are sixteen PMCON registers, one for each 256 Mbyte physical memory region on the 80960Hx (PMCON0 through PMCON15). There are fifteen pairs of LMAR/LMMR registers (LMAR0/LMMR0 through LMAR14/LMMR14), and one DLMCON register.

PMCON registers define the following physical memory attributes for the 80960Hx.

- Read Address to Data Wait States
- Read Data to Data Wait States
- Write Address to Data Wait States
- Write Data to Data Wait States
- Bus Turnaround Wait States
- Data Bus Parity
- Parity Sense
- Bus Width
- Read Pipelining
- Burst Enable
- $\overline{\text{READY}}/\overline{\text{BTERM}}$ Enable

Note that this is a superset of the control found in 80960Cx MCON registers. Also note that the format for the PMCON registers of the 80960Jx and 80960Hx match; 80960Jx PMCON registers select only bus width, but the bit positions are the same.

The PMCON register format is shown in Figure 11.

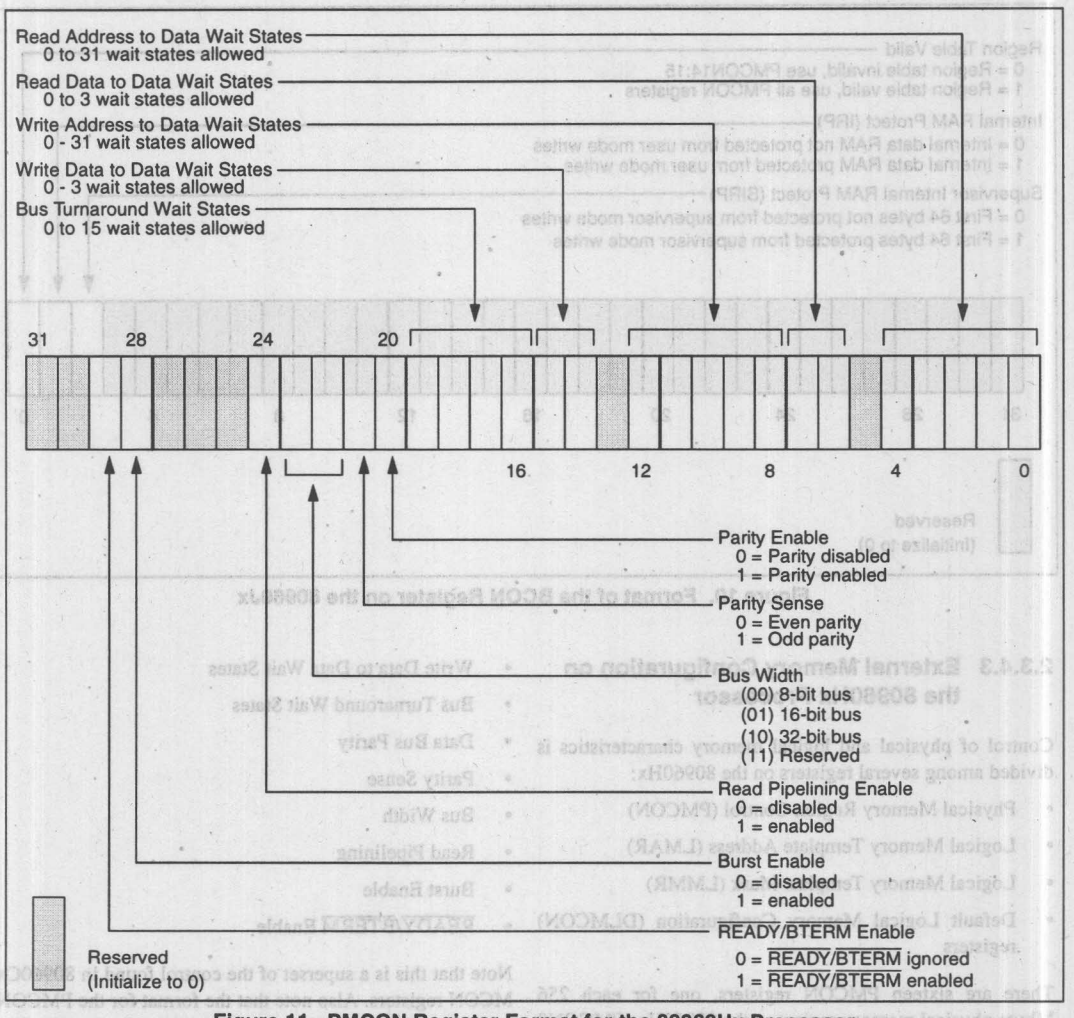


Figure 11. PMCON Register Format for the 80960Hx Processor

Logical memory control registers (LMARx and LMMRx) inform the processor of the logical attributes of external memory. On the 80960Hx, these registers control byte ordering, data caching, and regions to be marked for quick invalidation in the data cache. Lines in the data cache filled from a "quick invalidation" region may be invalidated by executing the Data Cache Control (**dcctl**) instruction, without invalidating the entire cache. Note that byte ordering is controlled by logical region on the 80960Hx; the 80960Jx supports only global byte ordering.

Logical memory regions are created using the LMAR/LMMR register pair. LMAR contains a starting address (4 Kbyte granularity), against which all external accesses are compared. LMMR contains a mask, which indicates which address bits in LMAR are actually compared (compare under mask). In this manner, one register pair may be used to create multiple logical regions, aliased on an arbitrary power-of-two boundary, by making some combination of upper memory bits "don't care." LMAR and LMMR register formats are shown in Figure 12.

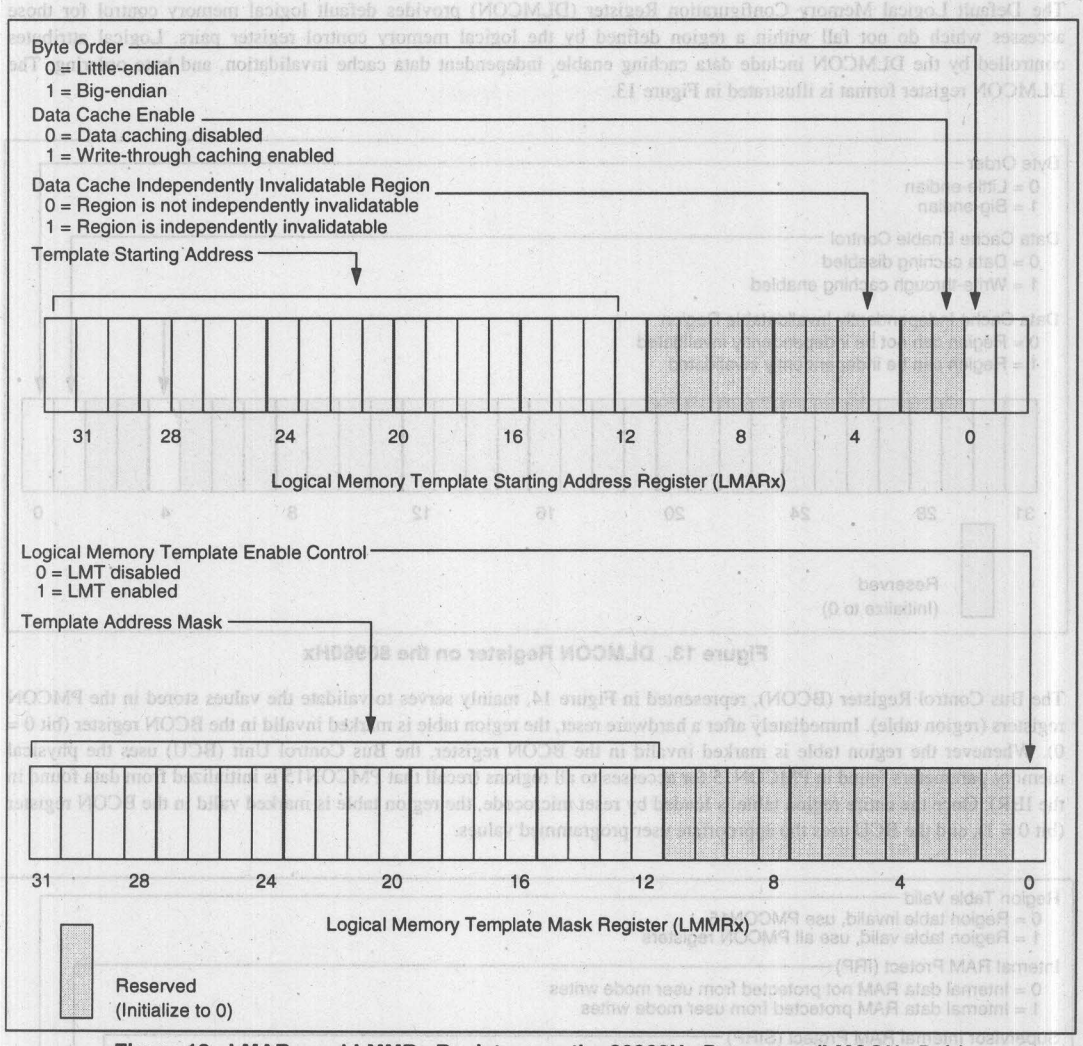


Figure 12. LMARx and LMMRx Registers on the 80960Hx Processor (LMCONx Pair)

The Default Logical Memory Configuration Register (DLMCON) provides default logical memory control for those accesses which do not fall within a region defined by the logical memory control register pairs. Logical attributes controlled by the DLMCON include data caching enable, independent data cache invalidation, and byte ordering. The DLMCON register format is illustrated in Figure 13.

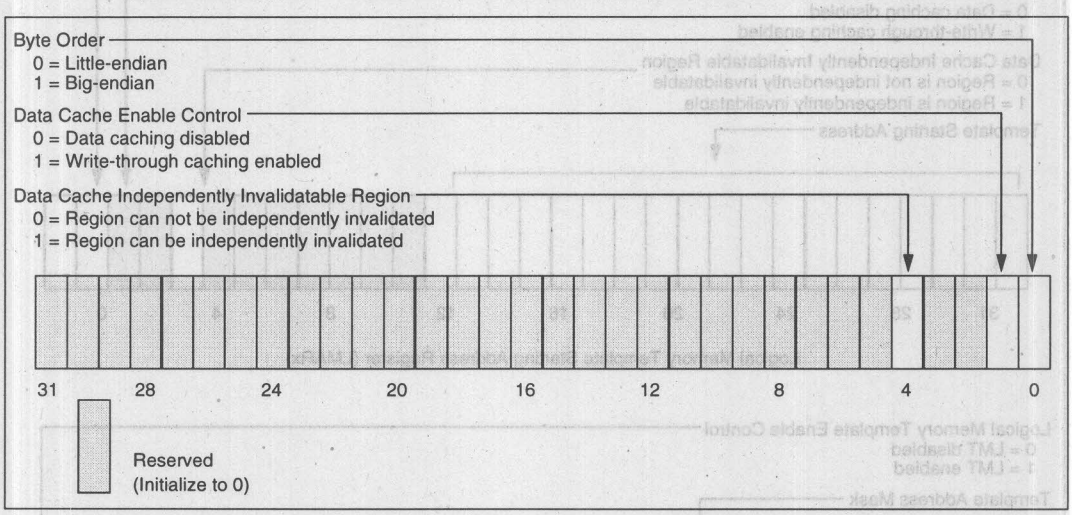


Figure 13. DLMCON Register on the 80960Hx

The Bus Control Register (BCON), represented in Figure 14, mainly serves to validate the values stored in the PMCON registers (region table). Immediately after a hardware reset, the region table is marked invalid in the BCON register (bit 0 = 0). Whenever the region table is marked invalid in the BCON register, the Bus Control Unit (BCU) uses the physical memory parameters found in PMCON15 for accesses to all regions (recall that PMCON15 is initialized from data found in the IBR). Once the entire region table is loaded by reset microcode, the region table is marked valid in the BCON register (bit 0 = 1), and the BCU uses the appropriate user programmed values.

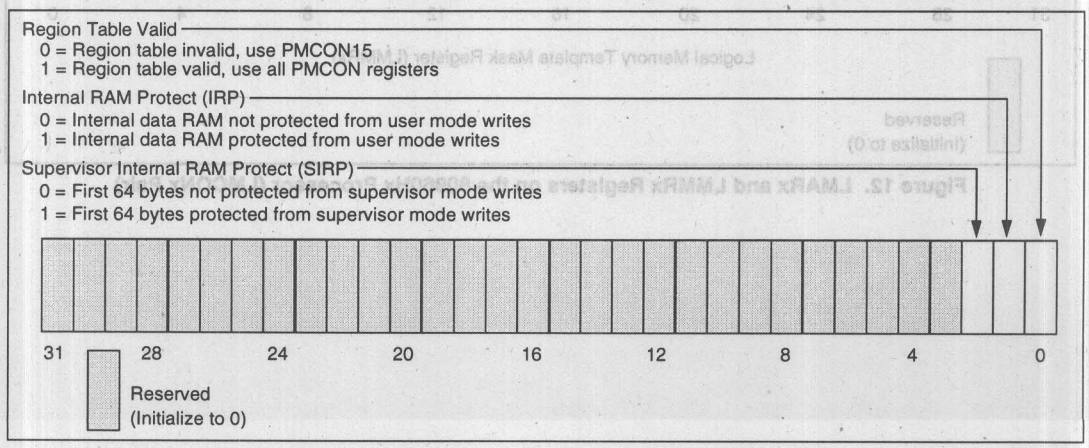


Figure 14. Format of the BCON Register on the 80960Hx

Table 6. New 80960Jx and 80960Hx Instructions

Instruction Name	Mnemonic	Processor	
		Jx	Hx
Conditional Integer/Ordinal Add	addi<cc>/addo<cc>	X	X
Conditional Integer/Ordinal Subtract	subi<cc>/subo<cc>	X	X
Byte Swap	bswap	X	X
Compare Integer/Ordinal Byte	cmpib/cmpob	X	X
Compare Integer/Ordinal Short	cmpis/cmpos	X	X
Select Value	sel<cc>	X	X
Data Cache Control	dcctl	X	X
Instruction Cache Control	icctl	X	X
Interrupt Control	intctl	X	X
Global Interrupt Disable	intdis	X	X
Global Interrupt Enable	inten	X	X
Halt	halt	X	X
Flush Data Cache Contents by Address	dcflusha		X
Give Address to Data Cache as Hint	dchint		X
Data Cache Invalidate by Address	dcinva		X

NOTES:

1. <cc> indicates conditional instructions. These conditions include: no - unordered, g - greater, e - equal, ge - greater or equal, l - less, le - less or equal, ne - not equal, o - ordered.

3.1.1 Conditional Integer/Ordinal Add and Subtract and Select Value Instructions

Conditional add instructions (**addi**<cc>/**addo**<cc>) provide an efficient method for encoding C language constructs. Consider the following example:

```
if (condition) value += adder; /* may translate to following assembly */
         cmpir4, 0/* condition in r4 */
         addiner5, r6, r5/* value in r5, adder in r6 */
```

This eliminates the need to branch around the addition. A similar example could be cited for the subtract instruction (**subi**<cc>/**subo**<cc>). The select value instruction (**sel**<cc>) provides the ability to conditionally move data, thus an efficient means to encode the following C construct:

```
r = a ? b : c; /* may translate to following assembly */
         cmpir4, 0/* a in r4, tested against 0000.0000H */
         seler5, r6, r7 /* r in r7, b in r5, c in r6 */
```

Again, this eliminates the need to branch around a move instruction.

The operation of these new instructions depends on the condition code mask applied by the instruction, to the current state of the condition code flags. Condition code masks are selected based on the <cc> field, and listed in Table 7.

Table 7. Condition Code Masks

Mask	Instruction	Mask	Instruction
000_2	Unordered	100_2	Less
001_2	Greater	101_2	Not Equal
010_2	Equal	110_2	Less or Equal
011_2	Greater or Equal	111_2	Ordered

NOTES:

1. Mask values correspond to those values applied to the condition code flags, found in the Arithmetic Controls register (AC.cc).

3.1.1.1 Operation of Conditional Add Instruction

The conditional add instructions perform the following operation:

```
if ((((mask != 000₂) && (mask & AC.cc)) != 000₂) || ((mask == 000₂) && (AC.cc == 000₂)))
 src/dst = src2 + src1;
```

3.1.1.2 Operation of Conditional Subtract Instruction

The conditional subtract instructions perform the following operation.

```
if ((((mask != 000₂) && (mask & AC.cc)) != 000₂) || ((mask == 000₂) && (AC.cc == 000₂)))
 src/dst = src2 - src1;
```

3.1.1.3 Operation of Select Value Instruction

The select value performs the following operation:

```
if ((((mask != 000₂) && (mask & AC.cc)) != 000₂) || ((mask == 000₂) && (AC.cc == 000₂)))
 src/dst = src2;
else
 src/dst = src1;
```

3.1.2 Byte Swap Instruction (bswap)

The byte swap instruction is useful for reversing the byte ordering of a value; it swaps a value's endianess from little- to big-endian, or vice versa. For instance, if a register contains FF00.AA55H, then after the **bswap** instruction, it will contain 55AA.00FFH. The operation of this instruction follows:

```
src/dst = ((src1 rotate 8) & 00FF.00FFH) +
((src1 rotate 24) & FF00.FF00H),
```

where rotate means to shift the value to the left by the number of bits specified; bits shifted out of the MSB position are inserted back at the LSB position. The user should be aware that, on some implementations, it is possible to perform the byte-swap operation using an equivalent set of discrete instructions.

3.1.3 Compare Integer/Ordinal Byte and Short Instructions

The compare integer/ordinal byte and short instructions provide for the comparison of 8- and 16-bit quantities, without the need to manually mask upper bits. The operation of these instructions follows:

```
if (src1[N:0] < src2[N:0]) AC.cc = 100₂;
if (src1[N:0] == src2[N:0]) AC.cc = 010₂;
if (src1[N:0] > src2[N:0]) AC.cc = 001₂;
```

where N indicates the number of least-significant bits to compare, and is either seven or fifteen.

3.1.4 Data Cache Control Instruction (dcctl)

The data cache control instruction performs data cache management and control. The *src1* value determines the operation to perform; some operations also use *src2* and *src/dst*. This instruction eliminates the need to provide such functionality in the **sysctl** instruction, thus improving its performance. The operations performed by this instruction are listed in the following sections.

3.1.4.1 Function Zero - Disable the Data Cache (*src1* == 0)

This function globally disables the data cache. Data loads and stores are directed to external memory; no access causes a hit or space to be allocated in the cache. The cache contents remain unchanged.

3.1.4.2 Function One - Enable the Data Cache (*src1* == 1)

This function globally enables the data cache. Data loads may hit in the cache, eliminating the external bus access. Data stores are allocated to the cache, and written to external memory (write through cache).

3.1.4.3 Function Two - Globally Invalidate the Data Cache (*src1* == 2)

This function invalidates the entire data cache.

3.1.4.4 Function Three - Ensure Coherency of the Data Cache with External Memory (*src1* == 3)

This function is meant to ensure that the data cache is guaranteed to be coherent with external memory. In this case, since the data cache is write-through, the cache is invalidated. In this manner, any new data allocated in the cache is coherent with external memory.

3.1.4.5 Function Four - Get Data Cache Status (*src1* == 4)

This function returns the status of the data cache. Status returned in *src/dst* follows the format outlined in Table 8.

Table 8. Format of Data Cache Status

Bits of *src/dst*	Meaning
[0]	Data Cache Enable: 0 = disabled, 1 = enabled
[7:4]	$\log_2$ (bytes per atom)
[11:8]	$\log_2$ (atoms per line)
[15:12]	$\log_2$ (number of sets)
[27:16]	Number of ways minus one

NOTES:

1. Total Cache Size = ([27:16] + 1) << ([7:4] + [11:8] + [15:12])
2. Bits not specified read as zeroes.

3.1.4.6 Function Six - Store Data Cache Sets to Memory (*src1* == 6)

This instruction provides a means to dump the entire data cache, so that its contents may be examined and evaluated. The operation performed by this function follows.

Table 9. Function Six - Store Data Cache Sets to Memory (*src1* == 6)

```
start = src/dst[15:0]; /* starting set number */
end = src/dst[31:16]; /*ending set number */
if (end >= DCACHE_MAX_SETS)
  end = DCACHE_MAX_SETS - 1;
if (start > end)
  GenerateFault(Operation.InvalidOperand);
memadr = src2;
for (set = start; set <= end; set++)
{
  memory[memadr] = SetData[set];
  memadr += 4;
  for (way = 0; way < DCACHE_MAX_WAYS; way++)
  {
    memory[memadr] = tags[set][way];
    memadr += 4;
    memory[memadr] = valid_bits[set][way];
    memadr += 4;
    for (word = 0; word < DCACHE_WORDS_IN_LINE; word++)
    {
      memory[memadr] = DCache_Line[set][way][word];
      memadr += 4;
    }
  }
}
```

3.1.5 Instruction Cache Control Instruction (icctl)

The instruction cache control instruction performs instruction cache management and control. The *sr1* value determines the operation to perform; some operations also use *src2* and *src/dst*. This instruction eliminates the need to provide such functionality in the **sysctl** instruction, thus improving its performance. The operations performed by this instruction are listed in the following subsections.

3.1.5.1 Function Zero - Disable the Instruction Cache (*src*1 == 0)

This function globally disables the instruction cache. Instruction fetches are directed to external memory; no fetch causes a hit or space to be allocated in the cache. The cache contents remain unchanged.

3.1.5.2 Function One - Enable the Instruction Cache (*src1* == 1)

This function globally enables the instruction cache. Instruction fetches may hit in the cache, eliminating the external bus access. Instruction fetches that do not hit are redirected to external memory, and allocated in the cache.

3.1.5.3 Function Two - Globally Invalidate the Instruction Cache (*src1* == 2)

This function invalidates the entire instruction cache.

3.1.5.4 Function Three - Load and Lock Code into the Instruction Cache (*src1* == 3)

This function is used to load and lock instructions into the instruction cache. *Src2* contains the number of contiguous blocks to load; *src/dst* contains the starting address of code to lock. The operation of this function follows:

Table 10. Function Three - Load and Lock Code into the Instruction Cache (*src1* == 3)

```
for (j = 0; j < src2; j++)
{
  way = way_associated_with_block(j);
  start = src/dst + j * block_size;
  end = start + block_size;
  for (i = start; i < end; i += 4)
  {
    set = set_associated_with(i);
    word = word_associated_with(i);
    ICache_Line[set][way][word] = memory[i];
    UpdateTagNValidBits(set, way, word);
    LockICache(set, way, word);
  }
}
```

3.1.5.5 Function Four - Get Instruction Cache Status (*src1* == 4)

This function returns the status of the data cache. Status returned in *src/dst* follows the format outlined in Table 11.

Table 11. Format of Instruction Cache Status

Bits of *src/dst*	Meaning
[0]	Instruction Cache Enable: 0 = disabled, 1 = enabled
[7:4]	$\log_2$ (bytes per atom)
[11:8]	$\log_2$ (atoms per line)
[15:12]	$\log_2$ (number of sets)
[27:16]	Number of ways minus one

NOTES:

1. Total Cache Size = ([27:16] + 1) << ([7:4] + [11:8] + [15:12])
2. Bits not specified read as zeroes.

3.1.5.6 Function Five - Get Instruction Cache Locking Status (*src1* == 5)

This function returns the instruction cache locking status. Status returned in *src/dst* follows the format outlined in Table 12.

Table 12. Format of Instruction Cache Locking Status

Bits of *src/dst*	Meaning
[7:0]	Number of blocks that lock
[23:8]	Block size in words
[31:24]	Number of locked blocks

3.1.5.7 Function Six - Store Instruction Cache Sets to Memory (*src1* == 6)

This instruction provides a means to dump the entire instruction cache, so that its contents may be examined and evaluated. The operation performed by this function follows.

Table 13. Function Six - Store Instruction Cache Sets to Memory (*src1* == 6)

```
start = src/dst[15:0]; /* starting set number */
end = src/dst[31:16]; /*ending set number */
if (end >= ICACHE_MAX_SETS)
  end = ICACHE_MAX_SETS - 1;
if (start > end)
  GenerateFault(Operation.InvalidOperand);
memadr = src2;
for (set = start; set <= end; set++)
{
  memory[memadr] = SetData[set];
  memadr += 4;
  for (way = 0; way < ICACHE_MAX_WAYS; way++)
  {
    memory[memadr] = tags[set][way];
    memadr += 4;
    memory[memadr] = valid_bits[set][way];
    memadr += 4;
    for (word = 0; word < ICACHE_WORDS_IN_LINE; word++)
    {
      memory[memadr] = ICache_Line[set][way][word];
      memadr += 4;
    }
  }
}
```

3.1.6 Interrupt Control Instruction

This instruction provides a means of globally enabling or disabling interrupts; the previous state of the interrupt controller is returned. The interrupt control instruction may also be used to interrogate the state of the interrupt controller, whether enabled or disabled. The processor guarantees that the new interrupt controller state is in effect before the instruction completes.

Executing this function globally enables interrupts by clearing the global interrupt enable bit of the Interrupt Control register (ICON.gie = 0). The previous state of ICON.gie is returned in bit zero of the *src/dst*.

3.1.6.3 Function Two - Return Interrupt Controller Status (*src1* = 2)

By executing this function, the state of the global interrupt enable bit of the Interrupt Control register (ICON.gie) is returned in bit zero of *src/dst*.

3.1.7 Global Interrupt Disable

This instruction provides a high-performance method for globally disabling interrupts. Interrupts are disabled by setting the global interrupt enable bit in the Interrupt Control register (ICON.gie = 1). In timing-critical routines, this instruction is preferable to the Interrupt Control instruction (**intctl**).

3.1.8 Global Interrupt Enable

This instruction provides a high-performance method for globally enabling interrupts. Interrupts are enabled by clearing the global interrupt enable bit in the Interrupt Control register (ICON.gie = 0). In timing-critical routines, this instruction is preferable to the Interrupt Control instruction (**intctl**).

3.1.9 Halt Instruction

The **halt** instruction provides a means for the processor to prepare itself to receive an interrupt, and to service that interrupt quickly. Once the **halt** instruction executes processing ceases until an interrupt is received of sufficient priority to cause an exit from the halted state. Such an bring the processor out of the halted state in this case.

3.1.9.2 Function One - Globally Enable Interrupts (*src1* = 1)

When this instruction executes with *src*1 equal to one, the global interrupt enable bit of the Interrupt Control register is cleared (ICON.gie = 0). This globally enables all interrupts; any interrupt of sufficient priority causes the processor to exit halted state in favor of servicing the interrupt.

3.1.9.3 Function Two - Halt Without Modifying the Current Interrupt State (*src1* = 2)

When this instruction executes with *src1* equal to two, the interrupt controller state remains unmodified. The global interrupt enable state of the processor is not affected.

3.1.10 Flush Data Cache Contents by Address Instruction (dcflusha)

If the effective address referenced by this instruction hits in the data cache, then steps must be taken to guarantee that the value cached for any part or all of the quad-word that the address specifies must be the same as the values for that quad-word in external memory. Since the 80960Hx incorporates a write-through cache, this instruction performs the same operation as the **dcinva** instruction; the quad-word of data in which the address falls is invalidated.

This instruction is available only on the 80960Hx processor.

3.1.11 Give Address to Data Cache as Hint Instruction (dchint)

This instruction is architecturally specified as either a hint to the data cache, or as a no-operation.

If implemented as a data cache hint, it generates an effective address that would be sent to the data cache, indicating that data at the specified address is likely to be needed soon. If this hint to the data cache results in a fault generation, the hint is ignored and no fault is generated.

This instruction is available only on the 80960Hx processor, and is implemented as a no-operation.

3.1.12 Data Cache Invalidate by Address Instruction (dcinva)

The effective address referenced in this instruction is sent to the data cache. The cache must guarantee that after execution of this instruction, the quad-word of data in which this effective address falls is not cached.

This instruction is available only on the 80960Hx processor.

3.2 Extended Instructions

The 80960Jx and 80960Hx processors extend the functionality of the **sysctl** instruction, as originally supported by the 80960Cx processor. This additional functionality supports access to memory-mapped control registers, and the breakpoint sharing mechanism.

3.2.1 System Control Instruction (sysctl)

The system control (**sysctl**) instruction present on the 80960Cx processor has expanded functionality on the 80960Jx and 80960Hx processors. The function that **sysctl** performs is determined from the message type of the instruction. The message type is obtained by logically ANDing *src1* with FF00H, and shifting the results to the right by eight bits ((*src1* & FF00H) >> 8). The various functions of the **sysctl** instruction are detailed in the following sections.

3.2.1.1 Function Zero - Post Software Interrupt (Message Type = 00H)

This function behaves identically to the 80960Cx processor. Typically, **sysctl** is used to request an interrupt in a program. The request interrupt message type (00H) is selected and the interrupt procedure pointer number is specified in the least significant byte of the instruction operand.

3.2.1.2 Function One - Invalidate the Instruction Cache (Message Type = 01H)

This function behaves identically to the 80960Cx processor.

3.2.1.3 Function Two - Configure the Instruction Cache (Message Type = 02H)

The 80960Cx processor supports four modes of operation: modes 0, 1, 4, and 6. The mode is determined by logically ANDing *src1* with FFH (*src1* & FFH). Mode 0 and mode 1 enable and disable the instruction cache, respectively. Mode 4 may be used to load and lock the entire instruction cache (1 Kbyte), while mode 6 is used to load and lock only half the instruction cache (512 bytes).

The 80960Jx and 80960Hx processors also support four modes of operation with slightly different behavior; they also are modes 0, 1, 4, and 6. Again modes 0 and 1 may be used to enable and disable the instruction cache, respectively. However, modes 4 and 6 differ slightly from the 80960Cx processor operation. These two modes are used to load and lock the instruction cache; the number of blocks (ways) to load and lock are indicated by *src2*. The algorithm used to load and lock the cache is the same as that outlined for the instruction cache control instruction (**icctl**).

3.2.1.4 Function Three - Software Reinitialization (Message Type 03H)

This function behaves identically to the 80960Cx processor. The 80960Jx and 80960Hx processors guarantee that the instruction and data caches are both invalidated and disabled before execution resumes.

3.2.1.5 Function Four - Load One Group of Control Registers (Message Type 04H)

This function is implemented on the 80960Cx processor. The 80960Jx and 80960Hx processors do not support this function.

3.2.1.6 Function Five - Modify One Memory-mapped Control Register (Message Type 05H)

This function provides a means to modify a specified memory-mapped control register. The upper sixteen bits of *src1* (*src1*[31:16]) contain the two low-order bytes of the address of the memory-mapped register to access. The new value to write is contained in *src2*, and *src3* contains a mask. After the operation completes, the previous value of the memory-mapped register is returned in *src/dst*. The operation performed by this function follows.

```
addr = (FF00H << 16) | (src1 >> 16);
temp = memory[addr];
memory[addr] = (src2 & src/dst)
| (temp & ~src/dst);
src/dst = temp;
```

This function is **not** supported on the 80960Cx processor.

3.2.1.7 Function Six - Breakpoint Resource Request (Message Type 06H)

To properly use the hardware breakpoint resources of the 80960Jx and 80960Hx, application code must first request access to these resources. To do so, the application must execute this function of the **sysctl** instruction, and use this function. After completion, the value of *src/dst* indicates the breakpoint resources available to the application. This is shown in Table 14. For more information on breakpoint resource sharing, refer to section 5.0, Breakpoint Resource Sharing Mechanism.

Table 14. *src/dst* Field Definitions for Breakpoint Resource Request

Bits of *src/dst*	Description
[3:0]	Number of available instruction breakpoint registers
[7:4]	Number of available data address breakpoint registers
[31:8]	Read as zeroes

This function is **not** supported on the 80960Cx processor.

4.0 Register Cache/Stack Frames

The 80960Cx, 80960Jx, and 80960Hx processors each implement a register cache to store local register sets during explicit and implicit calls. By definition, this memory is not a cache, but a FIFO; the last register set stored in the memory is the first to be retrieved. The register cache is configured from the Register Cache Configuration Word (RCCW), which is read from the PRCB during initialization (see section 2.1.2 Processor Control Block (PRCB)). This register's format and control functions of vary from processor to processor, as discussed in the following sections.

4.1 Register Cache Configuration on the 80960Cx Processor

The 80960Cx processor provides for up to fifteen local register sets to be stored on-chip. If more than five registers sets are specified, additional storage space is taken from internal data RAM, starting at the highest available location and working down from there. For instance, if storage is requested for six register sets, the register cache stores the first five, and data RAM locations 0000.03C0H through 0000.03FFH are used to store the sixth set (recall the 80960Cx provides 1 Kbyte of on-chip data RAM located from addresses 0000.0000H through 0000.03FFH). The user code initializes the least-significant four bits of the RCCW with the total number of register sets to be cached on-chip. If greater than five sets are requested, the required amount of on-chip data RAM is used as described above. The format of the RCCW is shown in Figure 15.

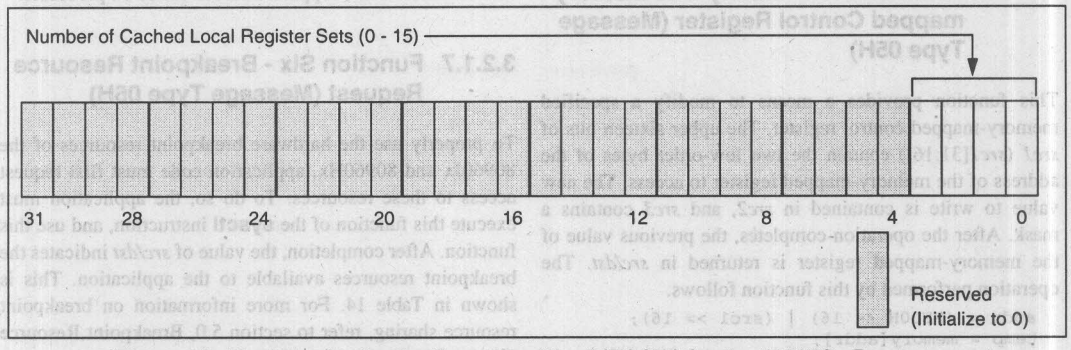


Figure 15. Register Cache Configuration Word (RCCW) for the 80960Cx Processor

4.2 Register Cache Configuration on the 80960Jx Processor

The 80960Jx processor provides for eight local register sets to be stored in the register cache. Unlike the 80960Cx and 80960Hx processors, additional register frames cannot be stored in the register cache by consuming additional on-chip data RAM.

By initializing the RCCW, application code may reserve register frames within the register cache for use only by high-priority interrupts (priority 28 or above). This is accomplished by programming bits 10 through 8 with the appropriate value. Allowed values of the programmed limit range from 000_2 to 111_2. Setting the value to 111_2 reserves all register frames for high-priority interrupts. Setting the value to 000_2 reserves no register frames for high-priority interrupts.

The RCCW is initialized from the PRCB during initialization. It is not accessible through the memory-mapped control register interface. Register format is shown in Figure 16.

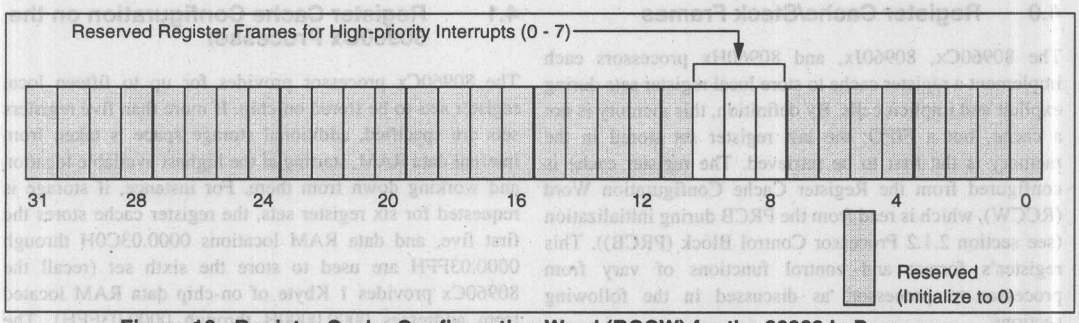


Figure 16. Register Cache Configuration Word (RCCW) for the 80960Jx Processor

4.3 Register Cache Configuration on the 80960Hx Processor

The 80960Hx processor provides for up to fifteen local register sets to be stored on-chip. If more than five register sets are specified, additional storage space is taken from internal data RAM, starting at the highest available location and working down from there. For instance, if storage is requested for six register sets, the register cache stores the first five, and data RAM locations 0000.07C0H through 0000.07FFH are used to store the sixth set (recall the 80960Hx provides 2 Kbytes of on-chip data RAM located from addresses 0000.0000H through 0000.07FFH).

User code initializes the least-significant four bits of the RCCW with the total number of register sets to be cached on-chip. If greater than five sets are requested, the required amount of on-chip data RAM is used as described above. By initializing

the RCCW appropriately, application code may reserve register frames within the register cache for use only for high-priority interrupts (priority 28 or above). This is accomplished by programming bits 12 through 8 with the appropriate value. Allowed values of the programmed limit range from 0000_2 to 1111_2. Setting the value to 1111_2 reserves all register frames for high-priority interrupts. Setting the value to 0000_2 reserves no register frames for high-priority interrupts. The number of frames reserved for high-priority interrupts must be equal to or less than the total number of frames available.

The RCCW is initialized from the PRCB during initialization. It is not accessible through the memory-mapped control register interface. The register format is shown in Figure 17.

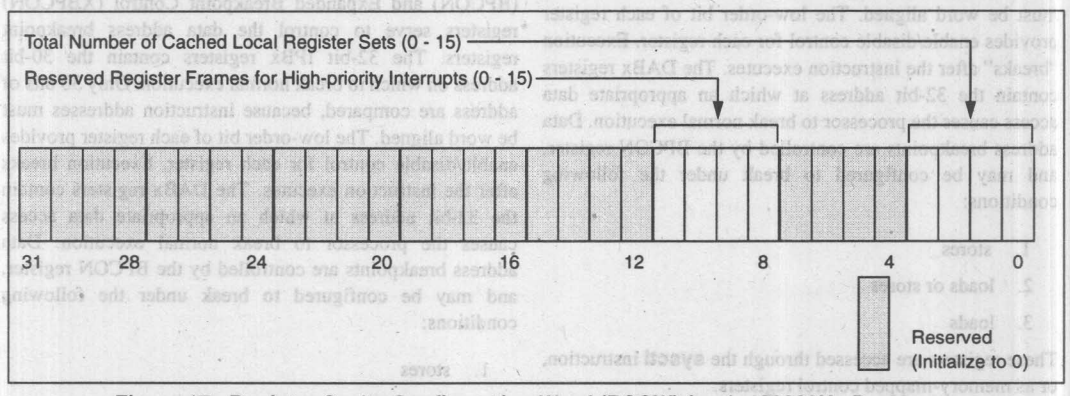


Figure 17. Register Cache Configuration Word (RCCW) for the 80960Hx Processor

5.0 Breakpoint Resource Sharing Mechanism

The 80960 family of processors provide significant on-chip debugging capabilities, including breakpoint and breakpoint control registers. While breakpoint resources included on the 80960Cx, 80960Jx, and 80960Hx processors are similar in function, the access model is different. Accesses to breakpoint resources on the 80960Jx and 80960Hx processors are controlled such that contention between application code (such as embedded debuggers) and hardware development tools does not occur. This section discusses these differences.

5.1 Breakpoint Resources on the 80960Cx Processor

The 80960Cx processor provides two instruction address (IPB0, IPB1) and two data address (DAB0, DAB1) breakpoint registers. In addition, the Breakpoint Control (BPCON) register serves to control the data address breakpoint registers. The 32-bit IPBx registers contain the 30-bit address on which to break normal execution. Only 30 bits of address are compared, because instruction addresses must be word aligned. The low-order bit of each register provides enable/disable control for each register. Execution "breaks" after the instruction executes. The DABx registers contain the 32-bit address at which an appropriate data access causes the processor to "break" normal execution. Data address breakpoints are controlled by the BPCON register, and may be configured to break on:

1. stores
2. loads or stores
3. load, store, or instruction fetch
4. any access

Application code programs these registers by using the "load control register" function of the **sysctl** instruction. The only restriction here is that the processor must be in supervisor mode to execute the **sysctl** instruction. These registers are also initialized at reset and software reinitialization, from values contained in the control table. There is no controlled access or sharing mechanism provided with the 80960Cx processor.

5.2 Breakpoint Resources on the 80960Jx Processor

The 80960Jx processor provides two instruction address (IPB0, IPB1) and two data address (DAB0, DAB1) breakpoint registers. In addition, the Breakpoint Control (BPCON) register serves to control the data address breakpoint registers. The 32-bit IPBx registers contain the 30-bit address on which to break normal execution. Only 30 bits of address are compared, because instruction addresses must be word aligned. The low-order bit of each register provides enable/disable control for each register. Execution "breaks" after the instruction executes. The DABx registers contain the 32-bit address at which an appropriate data access causes the processor to break normal execution. Data address breakpoints are controlled by the BPCON register, and may be configured to break under the following conditions:

1. stores
2. loads or stores
3. loads

These registers are accessed through the **sysctl** instruction, or as memory-mapped control registers.

To avoid contention between hardware development tools and application code, the 80960Jx implements a breakpoint sharing mechanism. Application code must always first request and acquire hardware breakpoint resources before any attempt is made to access them. Hardware development tools exercise supervisor control over breakpoint resource allocation. If the tool retains control of breakpoint resources, none are available for application code. If a development tool is not being used, access privilege to access breakpoint resources is granted to the application. The development tool may relinquish control of breakpoint resources to the application at its discretion.

If the application attempts to access the breakpoint or breakpoint control (BPCON) registers without first acquiring access rights to them, a fault is generated. In this case, the breakpoint resource is not modified, whether accessed through a **sysctl** instruction or as a memory-mapped register. These breakpoint registers are not initialized from the control table at reset or software reinitialization. Instead, they are initialized during the execution of the **sysctl** instruction when access is requested, only when the application successfully gains access to these registers.

For more information on breakpoint resource sharing, refer to the *i960® Jx Microprocessor User's Manual*.

5.3 Breakpoint Resources on the 80960Hx Processor

The 80960Hx processor provides six instruction address (IPB0 - IPB5) and six data address (DAB0 - DAB5) breakpoint registers. In addition, the Breakpoint Control (BPCON) and Expanded Breakpoint Control (XBPCON) registers serve to control the data address breakpoint registers. The 32-bit IPBx registers contain the 30-bit address on which to break normal execution. Only 30 bits of address are compared, because instruction addresses must be word aligned. The low-order bit of each register provides enable/disable control for each register. Execution breaks after the instruction executes. The DABx registers contain the 32-bit address at which an appropriate data access causes the processor to break normal execution. Data address breakpoints are controlled by the BPCON register, and may be configured to break under the following conditions:

1. stores
2. loads or stores
3. loads

These registers are accessed through the **sysctl** instruction, or as memory-mapped control registers.

To avoid contention between hardware development tools and application code, the 80960Hx also implements a breakpoint sharing mechanism. Application code must always first request and acquire hardware breakpoint resources before any attempt is made to access them. Hardware development tools exercise supervisor control over breakpoint resource allocation. If the tool retains control of breakpoint resources, none are available for application code. If the tool is not being used, access privilege to access breakpoint resources are granted to the application. It is possible on the 80960Hx for the application to obtain legal access to IPB0, IPB1, DAB0, and DAB1, and for the remaining breakpoint resources to remain under control of the development tool. Like the 80960Jx, the tool may relinquish control of breakpoint resources to the application at its discretion.

If the application attempts to access the breakpoint or breakpoint control (BPCON) registers without first acquiring access rights to them, a fault is generated. In this case, the breakpoint resource is not modified, whether

accessed through a **sysctl** instruction or as a memory-mapped register. These breakpoint registers are not initialized from the control table at reset or software reinitialization. Instead, they are initialized during the execution of the **sysctl** instruction when access is requested, only when the application successfully gains access to these registers.

For more information on breakpoint resource sharing, refer to the *i960® Hx Microprocessor User's Manual.*

6.0 Integrated Peripherals

The integrated peripheral sets available on the 80960Cx, 80960Jx, and 80960Hx processors are summarized in Table 15.

Table 15. Integrated Peripherals on the 80960Cx, 80960Jx, and 80960Hx Processors

Peripheral	Processor		
	80960Cx	80960Jx	80960Hx
Direct Memory Access Controller	Yes	No	No
Interrupt Controller	Yes	Yes	Yes
Guarded Memory Unit	No	No	Yes
Timers	None	Two	Two

6.1 Direct Memory Access Control Unit on the 80960Cx Processor

The Direct Memory Access controller on the 80960Cx processor provides integrated DMA capability to the application. The DMA controller can manage four independent DMA channels concurrently. Each channel supports transfers between any combination of external memory and internal data RAM. There are two modes employed to accomplish DMA transfers:

- demand mode (synchronous)
- block mode (asynchronous)

Demand mode transfers typically move data between external devices and memory. Block mode transfers typically move blocks of data within memory.

The 80960Cx processor provides a 13-pin interface to manage DMA transfers. These pins include:

- DMA request (DREQx)
- DMA acknowledge (DACKx)
- End of Process/Terminal Count (EOPx/TCx), and
- DMA Access (DMA)

The 80960Cx DMA controller provides two means to transfer data: standard and fly-by transfers. During standard transfers, multiple bus cycles are generated until the transfer is complete. During fly-by, the processor uses idle time on the bus to "slip in" a DMA bus cycle. Transfers may occur between any combination of 8-, 16-, and 32-bit memories using byte, short, word, or quad-word transactions.

DMA functionality is controlled by programming the DMA Command (DMAC) register, and the DMA Control Word. A special instruction, the Set Up DMA (**sdma**) instruction is also used to configure DMA functionality.

Programming the DMA controller is not discussed in detail in this document. Using the DMA capabilities of the 80960Cx processor precludes an application from being 80960Hx-ready. The user must not use DMA functions of the 80960Cx processor if the design will be upgraded to the 80960Hx processor. For more information on the DMA controller, refer to the *i960® Cx Microprocessor User's Manual.*

6.2 Guarded Memory Unit on the 80960Hx Processor

The Guarded Memory Unit (GMU) provides memory protection capabilities to the 80960Hx processor. This is useful during software development to detect and protect against illegal memory accesses. The GMU does not degrade execution, or in any way affect correct program execution. This peripheral is available only on the 80960Hx processor.

Using the GMU, application code may set up ranges of addresses with programmable access privileges. These privileges include: user read, user write, user execute, supervisor read, supervisor write, and supervisor execute. The GMU provides two range types: *protection* and *detection* ranges. Illegal accesses which fall within a protected range are prevented from altering memory. The processor guarantees that the access does not result in an internal cache or data RAM access, or generate an external bus cycle. The `PROTECTION.BAD_ACCESS` fault is also generated. An unauthorized access which falls within a detection range is allowed to complete, but the same protection fault is generated. An example of how the GMU might be configured to provide memory protection is given in Table 16.

Table 16. Sample Application with Restricted Memory Partitions

Application Memory Partition	Supervisor		User		Execute	
	Read	Write	Read	Write	Read	Write
Application/User RAM			X	X		
User Mode Stack	X	X	X	X		
Interrupt, Supervisor, Fault Tables	X		X			
Interrupt Stack	X	X				
Supervisor Stack	X	X				
Kernel RAM	X	X				
Kernel Code and Application Code					X	X

NOTES:

1. An 'X' in a location indicates the access is allowed.

The GMU is configured using several control registers:

- GMU Control register (GCR)
- GMU Memory Protect Address register pairs (MPARx, MPMRx), and
- GMU Memory Violation Detection register pairs (MDUBx, MULBx)

Two memory protect address register pairs are provided, allowing for a minimum of two protection ranges to be selected. More than two protection ranges are possible because the protection range address pairs are implemented as a base address and mask. An arbitrary number of addresses may be aliased by setting the appropriate mask bits. A total of six memory violation detection ranges may be selected. These register pairs act as true range (upper and lower bounds) registers.

Programming the GMU is not discussed in detail in this document. For more information on the GMU, refer to the *i960® Hx Microprocessor User's Manual.*

6.3 Interrupt Control Unit

The 80960Cx, 80960Jx, and 80960Hx processors implement an integrated interrupt controller. The interrupt controllers present on the 80960Jx and 80960Hx are identical. The 80960Jx and 80960Hx implementations draft directly from the 80960Cx. The following sections detail differences between the 80960Cx and the 80960Jx/ 80960Hx interrupt controllers.

6.3.1 80960Cx Processor Interrupt Control Unit

The 80960Cx processor's interrupt controller is configured by programming the Interrupt Control (ICON), Interrupt Mapping (IMAP0 - IMAP2), Interrupt Mask (IMSK), and Interrupt Pending (IPND) registers. The formats of these registers are given in the following paragraphs.

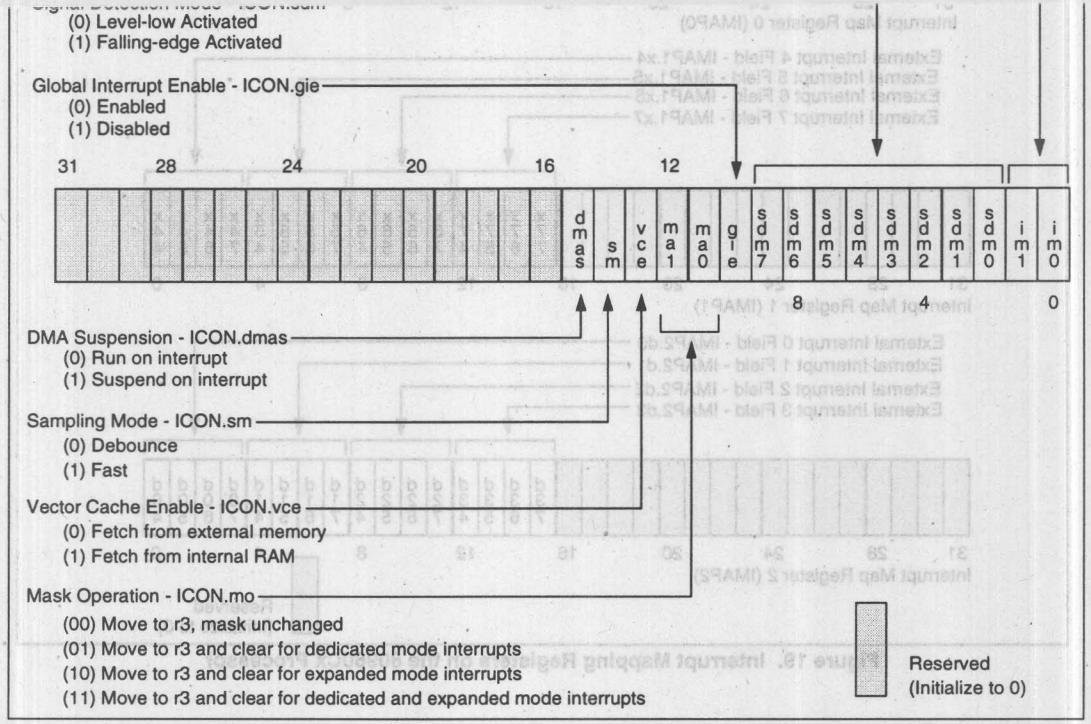


Figure 18. Interrupt Control Register on the 80960Cx Processor

Note the control bits designated in Figure 18 which are related to DMA functionality. The 80960Jx and 80960Hx processors do not support DMA capability, and therefore do not contain these control bits. Programming the ICON register is supported through the **sysctl** instruction on the 80960Cx processor.

6.3.1.2 Interrupt Mapping Registers (IMAP0-IMAP2)

Interrupt mapping registers are used to program the vector number associated with an interrupt source when the source is connected to a dedicated-mode input.

- Registers IMAP0 and IMAP1 contain mapping information for the external interrupt pins (four bits per pin
- Register IMAP2 contains mapping information for the DMA interrupt inputs (four bits per input)

The format of these registers is shown in Figure 19.

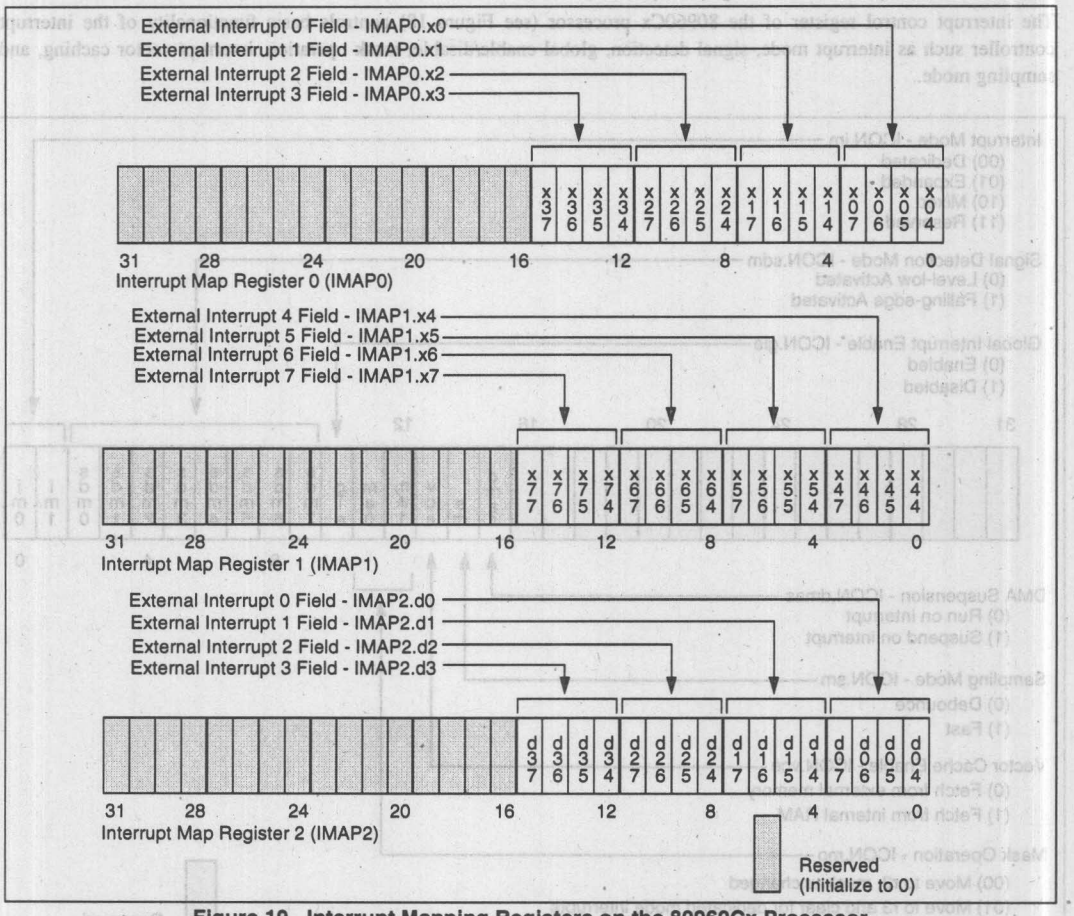


Figure 19. Interrupt Mapping Registers on the 80960Cx Processor

Note the DMA-related control bits. The 80960Jx and 80960Hx processors do not support DMA functionality; therefore these control bits do not exist. The IMAPx registers are programmed by using the **sysctl** instruction.

6.3.1.3 Interrupt Mask and Interrupt Pending Registers (IMSK, IPND)

The interrupt mask register provides a mechanism to uniquely mask dedicated mode interrupts and DMA interrupt sources, or globally mask expanded mode interrupts. In this manner, application software may prevent servicing of these interrupts. The interrupt pending register serves as a means to post dedicated mode interrupts, or interrupts from a DMA source. Application code which sets a bit in the interrupt pending register precipitates the same action that would occur if an external interrupt source caused the bit to be set.

These registers are accessed as special function registers: interrupt pending as sf0, interrupt mask as sf1. On reset, the interrupt mask is cleared (= 0) and interrupt pending undefined. Application code is responsible for explicitly initializing these registers. Figure 20 shows the format of these registers.

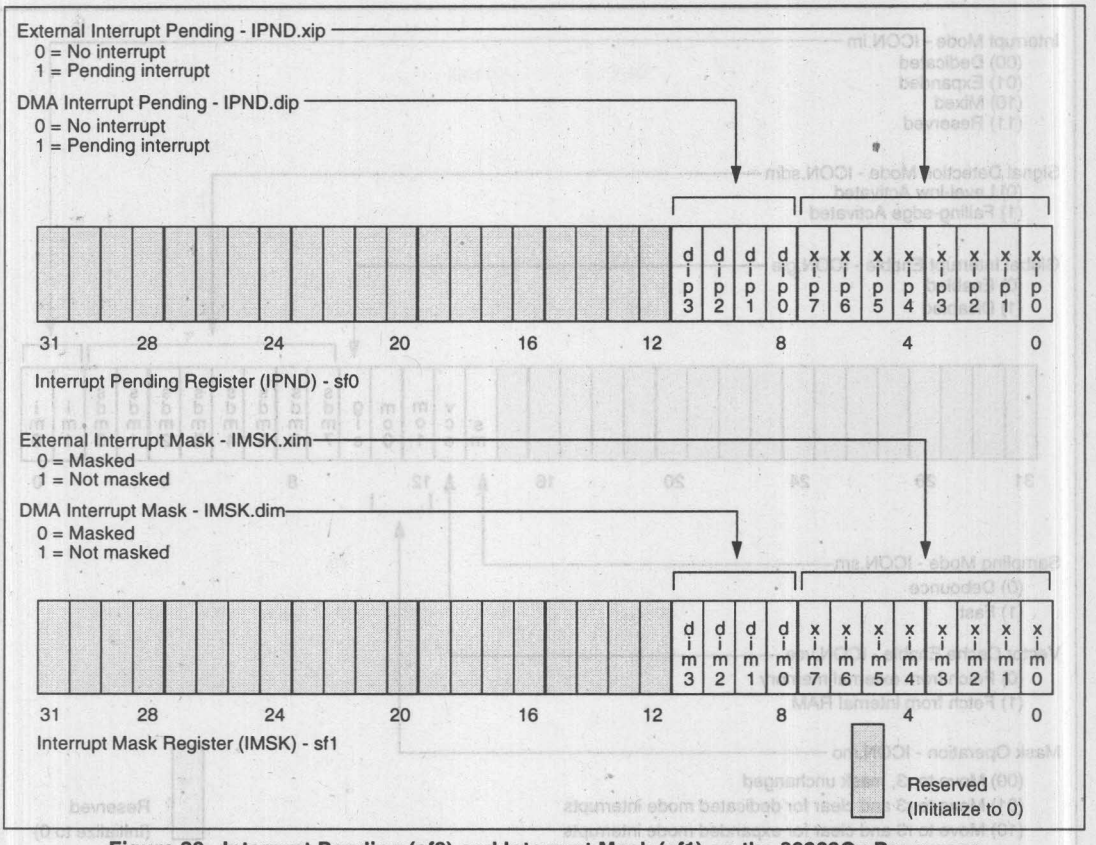


Figure 20. Interrupt Pending (sf0) and Interrupt Mask (sf1) on the 80960Cx Processor

6.3.2 80960Jx/80960Hx Processor Interrupt Control Unit

The 80960Jx and 80960Hx interrupt controllers are configured by programming the Interrupt Control (ICON), Interrupt Mapping (IMAP0 - IMAP2), Interrupt Mask (IMSK), and Interrupt Pending (IPND) registers. The formats of these registers are given in the following paragraphs.

6.3.2.1 Interrupt Control Register (ICON)

The interrupt control register of the 80960Jx and 80960Hx processors control basic functionality of the interrupt controller such as interrupt mode, signal detection, global enable/disable, mask operation, interrupt vector caching, and sampling mode. The format of this register is shown in Figure 21.

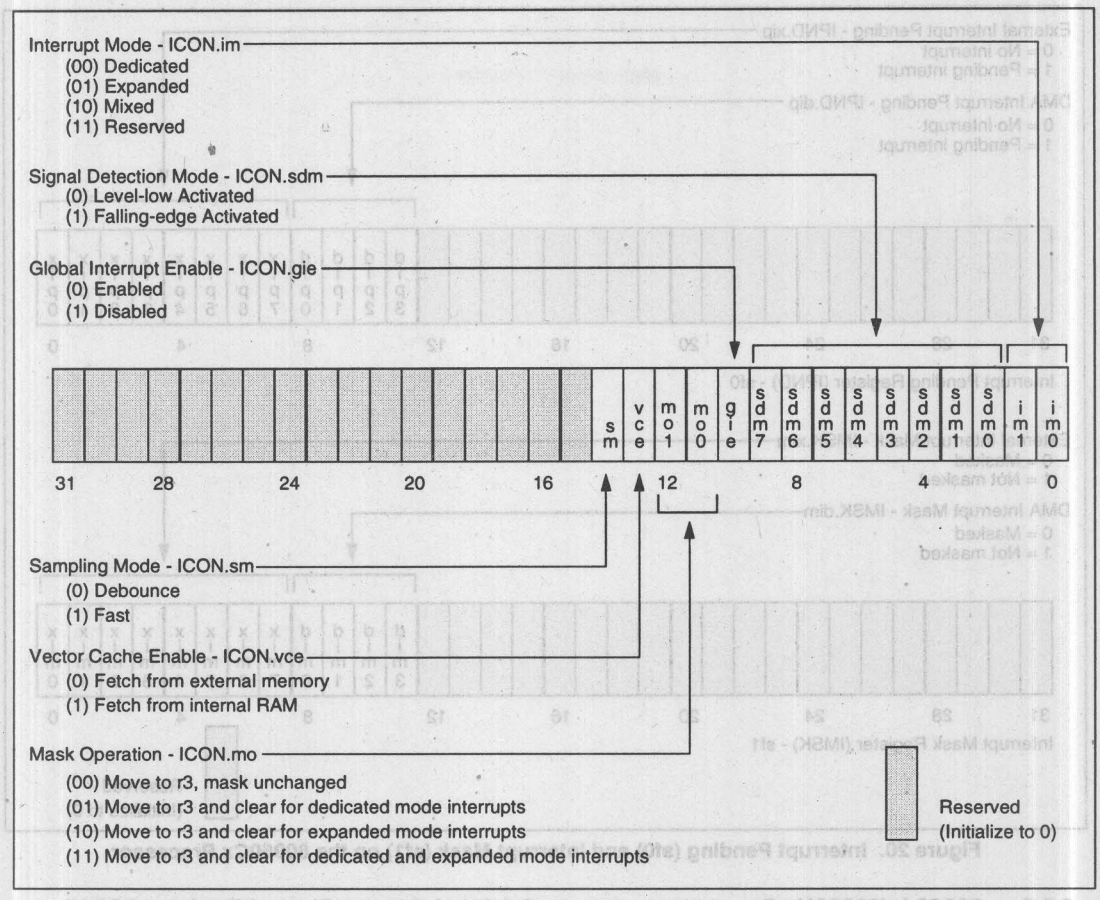


Figure 21. Interrupt Control Register (ICON) on the 80960Jx and 80960Hx Processors

Programming the ICON register is supported through the **sysctl** instruction. The ICON register is also mapped to the memory-mapped register space, address FF00.8510H, and may be accessed as such in supervisor mode. This register is also affected by the interrupt control (**intctl**), interrupt disable (**intdis**), and interrupt enable (**inten**) instructions. Refer to section 3.0 New and Extended Instructions, for more information.

6.3.2.2 Interrupt Mapping Registers (IMAP0-IMAP2)

Interrupt mapping registers are used to program the vector number associated with an interrupt source when the source is connected to a dedicated-mode input, or timer interrupt.

- Registers IMAP0 and IMAP1 contain mapping information for the external interrupt pins (four bits per pin)
- register IMAP2 contains mapping information for the timer interrupt sources (four bits per input)

The format of these registers is shown in Figure 22.

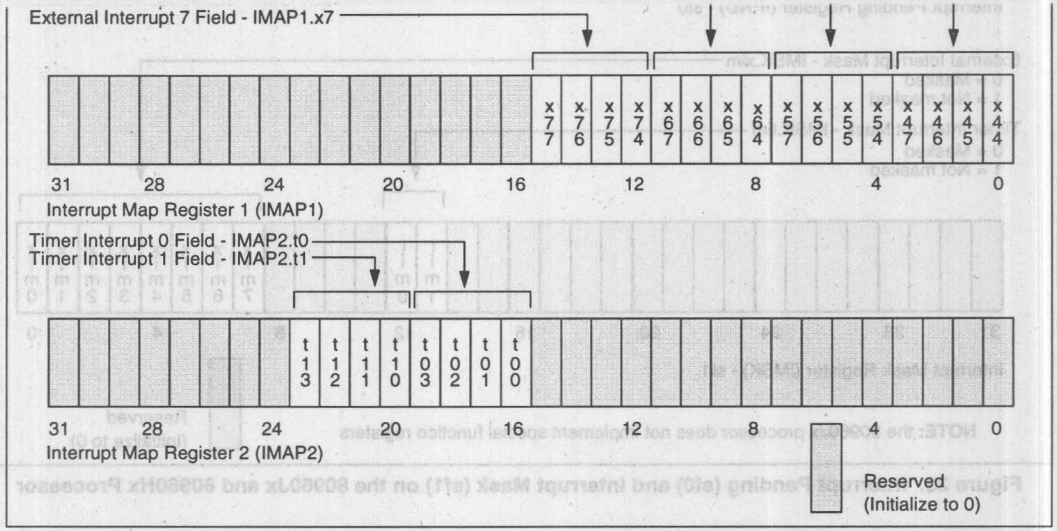


Figure 22. Interrupt Mapping Registers on the 80960Jx and 80960Hx Processors

The IMAPx registers are programmed by using the **sysctl** instruction. They may also be accessed as memory-mapped control registers, at addresses FF00.8520H through FF00.8528H, in supervisor-mode only.

6.3.2.3 Interrupt Mask and Interrupt Pending Registers (IMSK, IPND)

The interrupt mask register provides a mechanism to uniquely mask dedicated mode interrupts and timer interrupt sources, or globally mask expanded mode interrupts. In this manner, application software may prevent the servicing of these interrupts. The interrupt pending register serves as a means to post dedicated mode interrupts, or interrupts from a timer source. Application code which sets a bit in the interrupt pending register precipitates the same action that would occur if an external interrupt source caused the bit to be set.

These registers are accessed as special function registers: interrupt pending as sf0, interrupt mask as sf1. Note that the 80960Jx does not support special function registers. IMSK and IPND may be accessed as memory-mapped control registers, at address FF00.8504H and FF00.8500H, respectively. To access these memory-mapped registers, the application must be operating in supervisor mode. The **atmod**, **sysctl**, **ld**, and **st** instructions may also be used to modify these registers, again in supervisor-mode only.

On reset, interrupt mask is cleared (=0) and interrupt pending is undefined. Application code is responsible for explicitly initializing these registers. Figure 23 shows the format of these registers.

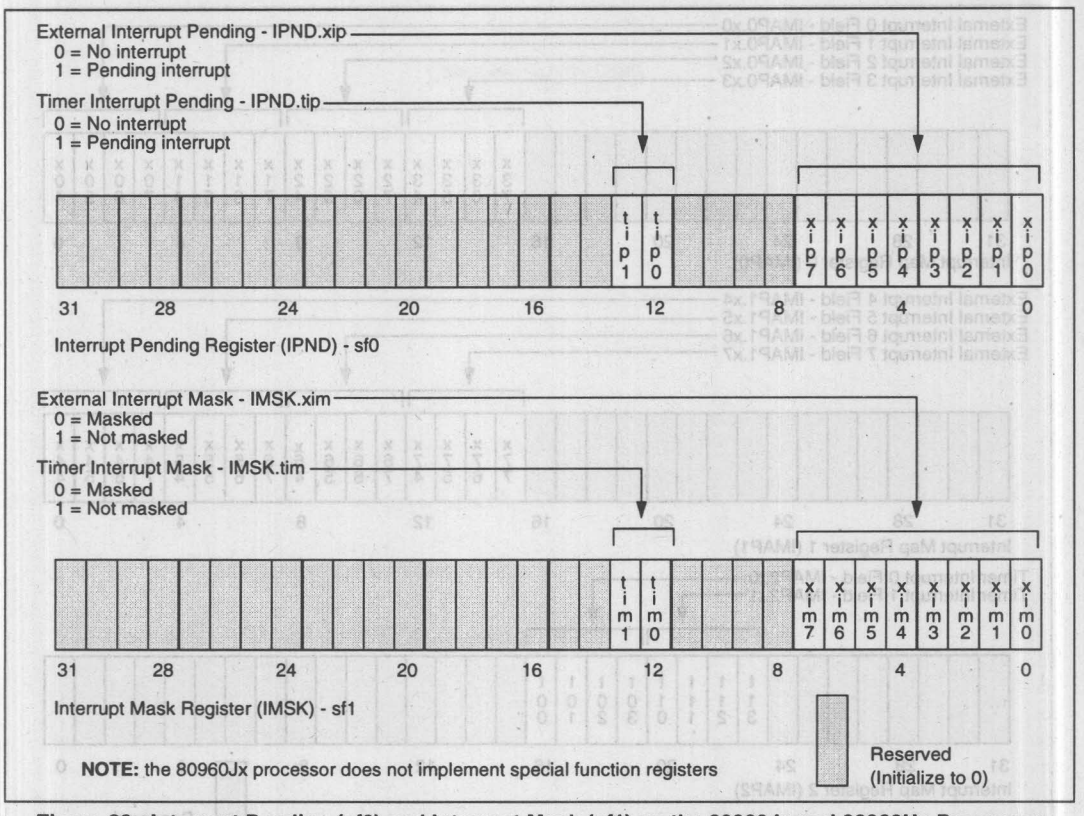


Figure 23. Interrupt Pending (sf0) and Interrupt Mask (sf1) on the 80960Jx and 80960Hx Processor

6.3.2.4 Improvements to Interrupt Latency

The 80960Jx and 80960Hx processors provide the ability to reserve local register frames in the on-chip register cache for use by interrupts of priority 28 or above. This mechanism minimizes the chances that high-priority interrupts result in a spill to external memory, potentially increasing interrupt latency. For more information on this topic, refer to Section 4.0, Register Cache/Stack Frames.

6.4 Timer Unit

The 80960Jx and 80960Hx processors integrate two general-purpose timers. Timers and timer controls are identical on the 80960Jx and 80960Hx processors. Refer to the *i960® Jx Microprocessor User's Manual* or the *i960® Hx Microprocessor User's Manual* for details. These 32-bit timers are useful as system clocks, or for other purposes. Each timer may be clocked independently (clock input based on CLKIN input). Timers operate in a single-shot mode, or auto-reload for continuous tick generation. Each timer may independently interrupt the core upon terminal count. Timer resources may be configured such that user-mode accesses generate a fault.

Timer resources are independently controlled by six control registers. These registers, along with their memory-mapped control register addresses, are listed in Table 17.

Table 17. Timer Control Registers on the 80960Jx and 80960Hx Processors

Name	Description	Memory-Mapped Address (Hex)
TRR0	Timer Reload Register 0	FF00.0300H
TCR0	Timer Count Register 0	FF00.0304H
TMR0	Timer Mode Register 0	FF00.0308H
TRR1	Timer Reload Register 1	FF00.0310H
TCR1	Timer Count Register 1	FF00.0314H
TMR1	Timer Mode Register 1	FF00.0318H

The timer mode control register enables/disables the timer, enables auto-reload, provides for protection from user-mode writes to timer resources, and controls the input clock to its associated timer. Timers may be clocked by:

- the bus clock (CLKIN)
- bus clock / 2
- bus clock / 4, or
- bus clock / 8

The timer count register contains the timer's current count, and may be read or written. The timer reload register contains the value to load into the timer count register upon reaching terminal count (when auto-reload mode is enabled).

The 80960Cx processor does not contain any integrated timers.

7.0 Memory-mapped Control Registers

The 80960Jx and 80960Hx processors implement memory-mapped control registers within the architecturally reserved memory locations from FF00.0000H to FFFF.FFFFH. Accesses to this address space **never** cause an external bus cycle. These memory-mapped control registers provide convenient access to control registers. Additionally, they provide visibility to internally cached values, such as the Processor Control Block (PRCB) pointer, which is not architecturally visible on the 80960Cx. **The 80960Cx processor *does not* implement memory-mapped control registers.**

Memory-mapped control registers residing in address locations FF00.0000H to FF00.7FFFH can be accessed in user or supervisor mode. Those located in addresses FF00.8000H to FFFF.FFFFH are accessible via supervisor mode only. When using load (**ld**) and store (**st**) instructions, application code must access memory-mapped registers as aligned words only. Unaligned and/or non-word accesses to these registers, or user mode accesses made to supervisor-only registers generate a fault. The **sysctl** instruction may also be used to atomically modify memory-mapped control registers. In addition, the **atmod** instruction may be used to atomically modify the Interrupt Pending (IPND) and Interrupt Mask (IMSK) registers.

Memory-mapped control register addresses are detailed in Section 8.0, Memory-mapped Control Register Address Space.

7.1 Special Function Registers and Memory-mapped Counterparts

The 80960Cx and 80960Hx processors implement special function registers (SFRs) to provide control over integrated peripherals. SFRs may be accessed by most REG format instructions. With the exception of the Cache Control Register, all of the 80960Hx SFRs are also accessible as memory-mapped control registers. **The 80960Jx *does not* implement special function registers.**

7.1.1 Special Function Registers on the 80960Cx Processor

The 80960Cx processor implements three special function registers:

- Interrupt Pending (sf0)
- Interrupt Mask (sf1), and
- DMA Command (sf2)

The formats of the Interrupt Pending and Interrupt Mask registers for the 80960Cx processor are shown in Section 6.3.1, 80960Cx Processor Interrupt Control Unit. DMA functionality is present only on the 80960Cx processor, and the DMA Command register as such is present only on the 80960Cx processor. The upper two bits (bits 30 and 31) of the DMA Command register control the data cache (CF processor only). For more information on DMA support, refer to the *i960® Cx Microprocessor User's Manual.*

7.1.2 Special Function Registers on the 80960Hx Processor

The 80960Hx processor implements five special function registers. With the exception of the data cache control register (sf2), these registers are also accessible as memory-mapped control registers. These registers, with their associated memory-mapped locations are listed in Table 18.

Table 18. Special Function Registers of the 80960Hx Processor

Name	Mnemonic	Special Function Register	Memory-mapped Address
Interrupt Pending Register	IPND	sf0	FF00.8500H
Interrupt Mask Register	IMSK	sf1	FF00.8504H
Data Cache Control Register	D	sf2	NA
Interrupt Control Register	INTCON	sf3	FF00.8510H
GMU Control Register		sf4	FF00.8000H

These registers may be accessed by application code as either special function or memory-mapped registers. However, these registers must not be accessed as both special function and memory-mapped registers within a short period of time (less than 64 core cycles). Doing so results in unpredictable behavior.

The formats of the interrupt pending, interrupt mask, and interrupt control registers are shown in Section 6.3.2, 80960Jx/80960Hx Processor Interrupt Control Unit. The format of the GMU Control Register is shown in Section 6.2, Guarded Memory Unit on the 80960Hx Processor, illustrates the format of the Data Cache Control Register.

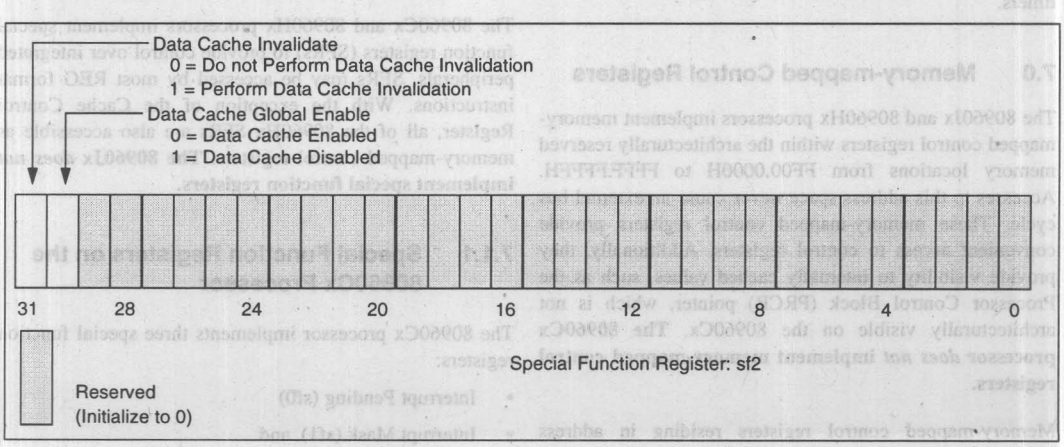


Figure 24. 80960Hx Processor Data Cache Control Register

The Data Cache Control register may be accessed as a special function register, or by use of the Data Cache Control instruction (**dcctl**). Regardless of how the register is accessed, bit 30 always properly reflects the state of the data cache: a zero if enabled and one if disabled. Writing a one to bit 31 causes the data cache to invalidate. Note that since the cache is write-through, it does not get flushed. When the invalidation process is complete, hardware automatically clears bit 31.

8.0 Memory-mapped Control Register Address Space

The memory-mapped control registers for the 80960Jx and 80960Hx microprocessors are shown in Table 19. Memory-mapped registers are not implemented on the 80960Cx processor. Accesses to this address space never causes an external bus cycle.

Table 19. Memory-mapped Control Registers (Sheet 1 of 5)

Address (Hex)	Description	Access Rights	Processor	
			Jx	Hx
FF00.0300H	Timer Reload Register 0 (TRR0)	S/U, R/W	X	X
FF00.0304H	Timer Count Register 0 (TCR0)	S/U, R/W	X	X
FF00.0308H	Timer Mode Register 0 (TMR0)	S/U, R/W	X	X
FF00.030CH	Reserved			
FF00.0310H	Timer Reload Register 1 (TRR1)	S/U, R/W	X	X
FF00.0314H	Timer Count Register 1 (TCR1)	S/U, R/W	X	X
FF00.0318H	Timer Mode Register 1 (TMR1)	S/U, R/W	X	X
FF00.031CH to FF00.7FFFH	Reserved			
FF00.8000H	GMU Control Register (GMCR, sf4)	S, R/W		X
FF00.8004H	Reserved			
FF00.8008H	Reserved			
FF00.8010H	Memory Protection Address 0 (MPAR0)	S, R/W		X
FF00.8014H	Memory Protection Mask 0 (MPMR0)	S, R/W		X
FF00.8018H	Memory Protection Address 1 (MPAR1)	S, R/W		X
FF00.801CH	Memory Protection Mask 1 (MPMR1)	S, R/W		X
FF00.8020H to FF00.807FH	Reserved			
FF00.8080H	Memory Detect Upper Bounds 0 (MDUB0)	S, R/W		X
FF00.8084H	Memory Detect Lower Bounds 0 (MDLB0)	S, R/W		X
FF00.8088H	Memory Detect Upper Bounds 1 (MDUB1)	S, R/W		X
FF00.808CH	Memory Detect Lower Bounds 1 (MDLB1)	S, R/W		X
FF00.8090H	Memory Detect Upper Bounds 2 (MDUB2)	S, R/W		X
FF00.8094H	Memory Detect Lower Bounds 2 (MDLB2)	S, R/W		X
FF00.8098H	Memory Detect Upper Bounds 3 (MDUB3)	S, R/W		X

Table 19. Memory-mapped Control Registers (Sheet 2 of 5)

Address (Hex)	Description	Access Rights	Processor	
			Jx	Hx
FF00.809CH	Memory Detect Lower Bounds 3 (MDLB3)	S, R/W		X
FF00.80A0H	Memory Detect Upper Bounds 4 (MDUB4)	S, R/W		X
FF00.80A4H	Memory Detect Lower Bounds 4 (MDLB4)	S, R/W		X
FF00.80A8H	Memory Detect Upper Bounds 5 (MDUB5)	S, R/W		X
FF00.80ACH	Memory Detect Lower Bounds 5 (MDLB5)	S, R/W		X
FF00.80B0H to FF00.80FFH	Reserved			
FF00.80E8H to FF00.80FFH	GMU Testability Registers			X
FF00.8100H	Default Logical Memory Configuration Register (DLMCON)	S, R/W	X	X
FF00.8104H	Reserved			
FF00.8108H	Logical Memory Address Register 0 (LMAR0)	S, R/W	X	X
FF00.810CH	Logical Memory Mask Register 0 (LMMR0)	S, R/W	X	X
FF00.8110H	Logical Memory Address Register 1 (LMAR1)	S, R/W	X	X
FF00.8114H	Logical Memory Mask Register 1 (LMMR1)	S, R/W	X	X
FF00.8118H	Logical Memory Address Register 2 (LMAR2)	S, R/W		X
FF00.811CH	Logical Memory Mask Register 2 (LMMR2)	S, R/W		X
FF00.8120H	Logical Memory Address Register 3 (LMAR3)	S, R/W		X
FF00.8124H	Logical Memory Mask Register 3 (LMMR3)	S, R/W		X
FF00.8128H	Logical Memory Address Register 4 (LMAR4)	S, R/W		X
FF00.812CH	Logical Memory Mask Register 4 (LMMR4)	S, R/W		X
FF00.8130H	Logical Memory Address Register 5 (LMAR5)	S, R/W		X
FF00.8134H	Logical Memory Mask Register 5 (LMMR5)	S, R/W		X
FF00.8138H	Logical Memory Address Register 6 (LMAR6)	S, R/W		X
FF00.813CH	Logical Memory Mask Register 6 (LMMR6)	S, R/W		X
FF00.8140H	Logical Memory Address Register 7 (LMAR7)	S, R/W		X
FF00.8144H	Logical Memory Mask Register 7 (LMMR7)	S, R/W		X
FF00.8148H	Logical Memory Address Register 8 (LMAR8)	S, R/W		X
FF00.814CH	Logical Memory Mask Register 8 (LMMR8)	S, R/W		X
FF00.8150H	Logical Memory Address Register 9 (LMAR9)	S, R/W		X
FF00.8154H	Logical Memory Mask Register 9 (LMMR9)	S, R/W		X
FF00.8158H	Logical Memory Address Register 10 (LMAR10)	S, R/W		X

Table 19. Memory-mapped Control Registers (Sheet 3 of 5)

Address (Hex)	Description	Access Rights	Processor	
			Jx	Hx
FF00.815CH	Logical Memory Mask Register 10 (LMMR10)	S, R/W		X
FF00.8160H	Logical Memory Address Register 11 (LMAR11)	S, R/W		X
FF00.8164H	Logical Memory Mask Register 11 (LMMR11)	S, R/W		X
FF00.8168H	Logical Memory Address Register 12 (LMAR12)	S, R/W		X
FF00.816CH	Logical Memory Mask Register 12 (LMMR12)	S, R/W		X
FF00.8170H	Logical Memory Address Register 13 (LMAR13)	S, R/W		X
FF00.8174H	Logical Memory Mask Register 13 (LMMR13)	S, R/W		X
FF00.8178H	Logical Memory Address Register 14 (LMAR14)	S, R/W		X
FF00.817CH	Logical Memory Mask Register 14 (LMMR14)	S, R/W		X
FF00.8180H to FF00.83FFH	Reserved			
FF00.8400H	Instruction Address Breakpoint 0 (IPB0)	S, Sys, WwG	X	X
FF00.8404H	Instruction address Breakpoint 1 (IPB1)	S, Sys, WwG	X	X
FF00.8408H	Instruction Address Breakpoint 2 (IPB2)	S, Sys, WwG		X
FF00.840CH	Instruction address Breakpoint 3 (IPB3)	S, Sys, WwG		X
FF00.8410H	Instruction Address Breakpoint 4 (IPB4)	S, Sys, WwG		X
FF00.8414H	Instruction address Breakpoint 5 (IPB5)	S, Sys, WwG		X
FF00.8418H to FF00.841FH	Reserved			
FF00.8420H	Data Address Breakpoint 0 (DAB0)	S, R/W, WwG	X	X
FF00.8424H	Data Address Breakpoint 1 (DAB1)	S, R/W, WwG	X	X
FF00.8428H	Data Address Breakpoint 2 (DAB2)	S, R/W, WwG		X
FF00.842CH	Data Address Breakpoint 3 (DAB3)	S, R/W, WwG		X
FF00.8430H	Data Address Breakpoint 4 (DAB4)	S, R/W, WwG		X
FF00.8434H	Data Address Breakpoint 5 (DAB5)	S, R/W, WwG		X
FF00.8438H to FF00.843FH	Reserved			
FF00.8440H	Breakpoint Control (BPCON)	S, R/W, WwG	X	X
FF00.8444H	Expanded Breakpoint Control (XBPCON)	S, R/W, WwG		X
FF00.8448H to FF00.84FFH	Reserved			
FF00.8500H	Interrupt Pending (IPND, sf0)	S, R/W, AtMod	X	X

Table 19. Memory-mapped Control Registers (Sheet 4 of 5)

Address (Hex)	Description	Access Rights	Processor	
			Jx	Hx
FF00.8504H	Interrupt Mask (IMSK, sf1)	S, R/W, AtMod	X	X
FF00.8508H to FF00.850FH	Reserved			
FF00.8510H	Interrupt Control (ICON, sf3)	S, R/W	X	X
FF00.8514H to FF00.851FH	Reserved			
FF00.8520H	Interrupt Map 0 (IMAP0)	S, R/W	X	X
FF00.8524H	Interrupt Map 1 (IMAP1)	S, R/W	X	X
FF00.8528H	Interrupt Map 2 (IMAP2)	S, R/W	X	X
FF00.852CH to FF0085FFH	Reserved			
FF00.8600H	80960Jx: Physical Memory Control Region 0:1 (PMCON0:1) 80960Hx: Physical Memory Control Region 0 (PMCON0)	S, R/W	X	X
FF00.8604H	Physical Memory Control Region 1 (PMCON1)	S, R/W		X
FF00.8608H	80960Jx: Physical Memory Control Region 2:3 (PMCON2:3) 80960Hx: Physical Memory Control Region 2 (PMCON2)	S, R/W	X	X
FF00.860CH	Physical Memory Control Region 3 (PMCON3)	S, R/W		X
FF00.8610H	80960Jx: Physical Memory Control Region 4:5 (PMCON4:5) 80960Hx: Physical Memory Control Region 4 (PMCON4)	S, R/W	X	X
FF00.8614H	Physical Memory Control Region 5 (PMCON5)	S, R/W		X
FF00.8618H	80960Jx: Physical Memory Control Region 6:7 (PMCON6:7) 80960Hx: Physical Memory Control Region 6 (PMCON6)	S, R/W	X	X
FF00.861CH	Physical Memory Control Region 7 (PMCON7)	S, R/W		X
FF00.8620H	80960Jx: Physical Memory Control Region 8:9 (PMCON8:9) 80960Hx: Physical Memory Control Region 8 (PMCON8)	S, R/W	X	X
FF00.8624H	Physical Memory Control Region 9 (PMCON9)	S, R/W		X
FF00.8628H	80960Jx: Physical Memory Control Region 10:11 (PMCON10:11) 80960Hx: Physical Memory Control Region 10 (PMCON10)	S, R/W	X	X

Table 19. Memory-mapped Control Registers (Sheet 5 of 5)

Address (Hex)	Description	Access Rights	Processor	
			Jx	Hx
FF00.862CH	Physical Memory Control Region 11 (PMCON11)	S, R/W		X
FF00.8630H	80960Jx: Physical Memory Control Region 12:13 (PMCON12:13) 80960Hx: Physical Memory Control Region 12 (PMCON12)	S, R/W	X	X
FF00.8634H	Physical Memory Control Region 13 (PMCON13)	S, R/W		X
FF00.8638H	80960Jx: Physical Memory Control Region 14:15 (PMCON14:15) 80960Hx: Physical Memory Control Region 14 (PMCON14)	S, R/W	X	X
FF00.863CH	Physical Memory Control Region 15 (PMCON15)	S, R/W		X
FF00.8640H to FF00.86FBH	Reserved			
FF00.86FCH	Bus Configuration Control (BCON)	S, R/W	X	X
FF00.8700H	Processor Control Block Pointer (PRCBptr)	S, RO	X	X
FF00.8704H	Interrupt Stack Pointer (ISP)	S, R/W	X	X
FF00.8708H	Supervisor Stack Pointer (SSP)	S, R/W	X	X
FF00.870CH	Reserved			
FF00.8710H	Device Identification (DEVICEID)	RO	X	X
FF00.8714H to FFFF.FFFFH	Reserved			

NOTES:

1. AtMod: Register can be updated quickly through the **atmod** instruction.
   RO: Read only.
   R/W: Register readable or writable.
   S: Supervisor access only.
   Spcl: Special Intel only test register.
   S/U: Supervisor or User access allowed, may be configured to be supervisor only.
   Sys: Modifiable using the **sysctl** instruction.
   WwG: Writing or modifying (with **st** or **sysctl**) the register only allowed when access has been granted.

Register locations which are not implemented are specifically reserved by Intel. Any locations not specifically defined are reserved by Intel.

Note that access to the instruction and data address breakpoint registers requires that the processor grant permission to access these registers. This is discussed in further detail in Section 5.0, Breakpoint Resource Sharing Mechanism.

9.0 80960Hx/80960Cx Pin Compatibility

Like the 80960Cx, the 80960Hx is available in the 168-lead PGA package. However, the 80960Hx is not 100% pin-compatible with the 80960Cx. This is largely due to the lack of Direct Memory Access (DMA) capability on the 80960Hx. On the 80960Cx, DMA capability consumed thirteen pins. The 80960Hx incorporates enhanced bus control functions and JTAG; pins required to support these features replace the DMA pins present on the 80960Cx. In this manner, it is possible to design a board that accepts the 80960Cx processor now, and the 80960Hx in the future. Pin diagrams for the 168-lead PGA package are illustrated in Figure 25 and Figure 26. See *Application Note 506, Designing for 80960Cx and 80960Hx Compatibility* for more details.

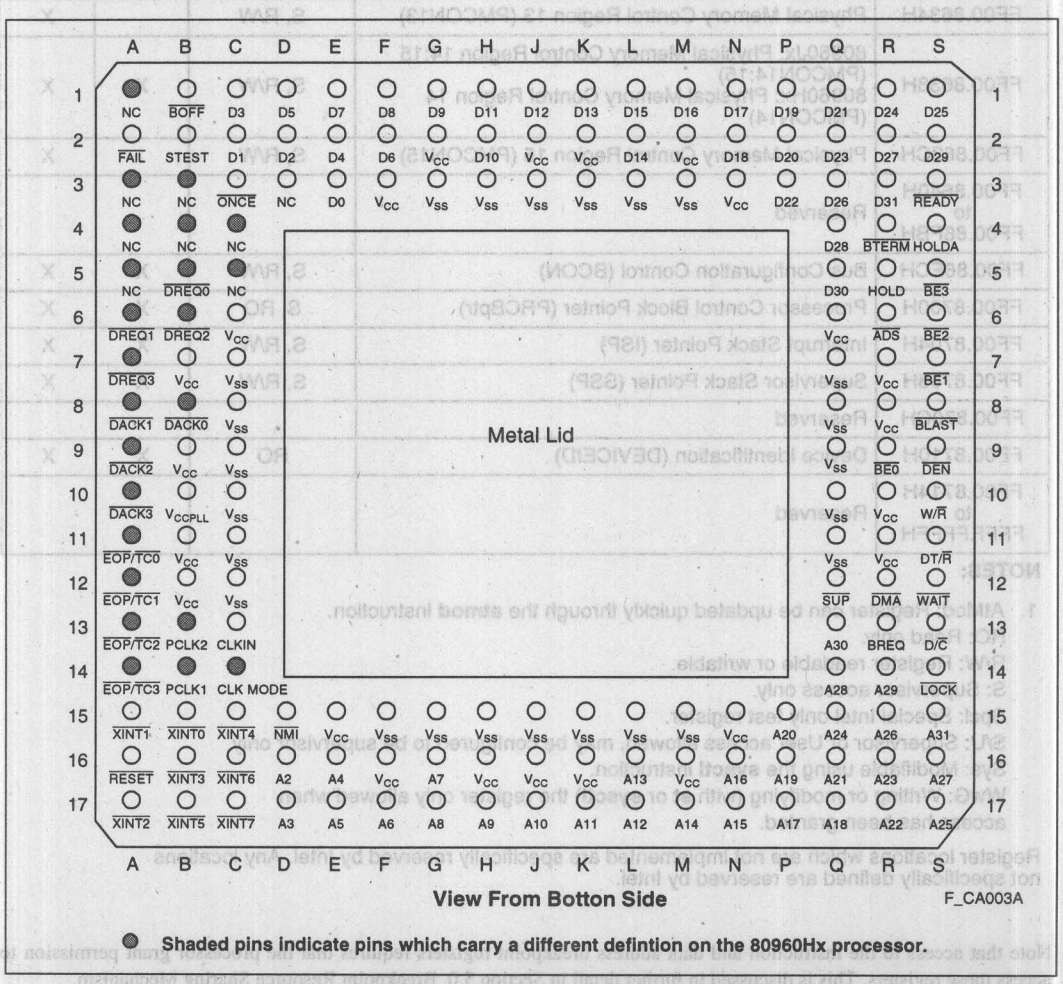


Figure 25. PGA Pinout Diagram for the 80960Cx Processor

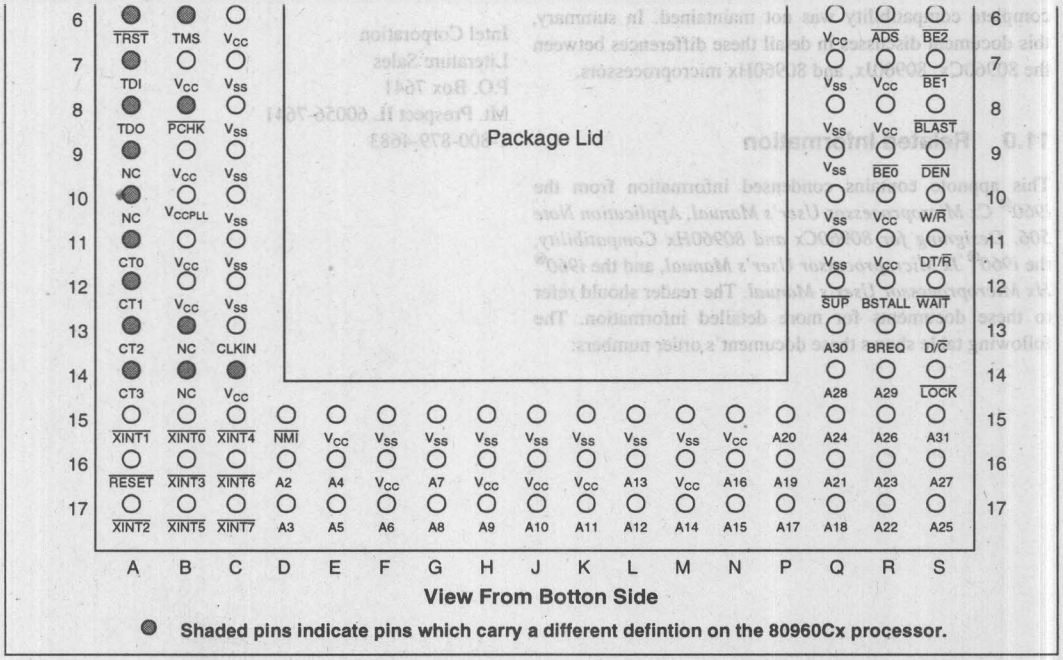


Figure 26. PGA Pinout Diagram for the 80960Hx Processor

The 80960Jx incorporates a multiplexed address and data bus. Therefore, pin compatibility with the 80960Cx or 80960Hx is not required or relevant.

10.0 Conclusion

In conclusion, this document describes three implementations of the i960® architecture: the 80960Cx, 80960Jx, and 80960Hx microprocessors.

During the 80960Jx and 80960Hx definition process, every attempt was made to maintain backward compatibility with the 80960Cx processor. To a large extent, this compatibility is maintained; most application code will run on each processor without modification. Due to enhancements to the bus interface, instruction set, and other refinements, complete compatibility was not maintained. In summary, this document discusses in detail these differences between the 80960Cx, 80960Jx, and 80960Hx microprocessors.

11.0 Related Information

This appnote contains condensed information from the *i960® Cx Microprocessor User's Manual*, *Application Note 506, Designing for 80960Cx and 80960Hx Compatibility*, the *i960® Jx Microprocessor User's Manual*, and the *i960® Hx Microprocessor User's Manual*. The reader should refer to these documents for more detailed information. The following table shows these document's order numbers:

Document Name	Order #
i960® Cx Microprocessor User's Manual	270710
i960® Jx Microprocessor User's Manual	272483
i960® Hx Microprocessor User's Manual	272484
Application Note 506—Designing for 80960Cx and 80960Hx Compatibility	272556

To receive these documents or any other available Intel literature, contact:

Intel Corporation
Literature Sales
P.O. Box 7641
Mt. Prospect IL 60056-7641
1-800-879-4683

intel

AP-726

APPLICATION NOTE

Interfacing the i960® Jx Microprocessor to the NEC µPD98401* Local ATM Segmentation and Reassembly (SAR) Chip

September, 21, 1995

Order Number: 272779-001

AP-726

APPLICATION NOTE

Interfacing the i960® Jx Microprocessor to the NEC μPD98401* Local ATM Segmentation and Reassembly (SAR) Chip

September, 21, 1995

Order Number: 272729-001

1.0 INTRODUCTION

This application note describes the interface between Intel's i960® JA/JF/JD microprocessors (referred to herein as the "i960 Jx processor") and NEC's µPD98401® Local ATM Segmentation And Reassembly (SAR) Chip, with 2 Mbytes of DRAM available to the i960 Jx processor. The interface provides for single and burst reads/writes to the DRAM and single reads and writes to the SAR chip, using the i960 Jx processor configured with a 32-bit bus. The interface is controlled by a DRAM controller and a SAR controller.

The DRAM interface features include:

- 32 bit wide bus, allowing 16 and 8 bit accesses
- Two-way interleaving
- 72 pin, 60 ns, 256 Kbyte x 32 bit Fast Page Mode DRAM SIMMs
- 1-0-0-0 wait state burst reads at 25 MHz
- 1-1-1-1 wait state burst writes at 25 MHz

The µPD98401 SAR chip interface features three wait state 32 bit reads and writes.

This document discusses DRAM controller theory with interleaving, and a basic slave read/write controller to the SAR. It also describes the supporting state machines, timing diagrams, and PLD equations. The µPD98401 can operate as a master device, which allows it access to DRAM. However, when the SAR is a master device, it can only request and obtain the i960 Jx processor bus, through the use of the processor's bus arbitration facilities. In this case, upon receiving HOLD from the SAR controller, the processor asserts HOLDA and three-states its signals, which grants bus control to the SAR.

1.1 Design Goals

The goal of this document is to provide a design aid for interfacing an i960 Jx processor to an NEC µPD98401 ATM SAR Chip. The application note defines an interface that was developed and tested on a i960 Jx processor-compliant platform, and can be plugged into a PCI expansion slot.

The platform used is the Intel Cyclone i960® Microprocessor PCI-SDK Evaluation Platform (referred to herein as the "PCI-SDK"), with an ATM Squall module and a i960 Jx processor CPU module.

Benefits of using the PCI-SDK include:

- The PCI-SDK plugs directly into a DOS-based system's PCI slot.
- While the user is working on their hardware design, code (software) for the design can be written concurrently, which saves development time.
- The design can be implemented on the PCI-SDK through a Squall module, which enhances testability.

1.2 Overview

The PCI-SDK consists of a base board with connectors which accept "modules" (daughter boards) for a CPU module and a Squall module. Many CPU and Squall modules are available; refer to Section 6.0, RELATED INFORMATION (pg. 6-20) for access to additional product information.

- The CPU modules are interchangeable daughter boards for Intel's family of i960 processors.
- The Squall modules are interchangeable daughter boards. Users can use these to design custom applications, or use one of PCI-SDK's existing modules, such as an Ethernet Squall II* module.

The PCI-SDK base board features include:

- Two SIMM sockets which support 2, 8, or 32 Mbytes of DRAM
- One Flash ROM socket
- Three 16-bit counter/timers or one 32-bit and one 16-bit counter
- DIP switch-selectable CPU clock frequency, for operation from 16 MHz to 50 MHz
- DRAM controller which automatically optimizes wait states to CPU frequency and memory speed
- One RS-232 serial port
- One Centronics compatible parallel download port
- PCI Bus Interface

The configuration used for development of this application note is:

CPU Module:	i960 Jx processor
CPU Bus Frequency:	25 MHz
DRAM:	2 banks of SIMMs: 60 ns, 2 Mbyte
Squall Module:	NEC µPD98401® - based ATM module

address cycle can be followed by up to four data cycles.

The DRAM SIMM has four $\overline{CAS}$ lines, one for each of eight (nine) bits in a 32-bit (36-bit) SIMM module. The four $\overline{CAS}$ lines control the writing to individual bytes within each SIMM.

1.4 DRAM Refresh

To maintain their data, the DRAM must be refreshed every 16 µs. This is achieved by a 6-bit refresh counter embedded in the DRAM controller, driven by a 4 MHz clock. The 4 MHz clock is created by dividing a 16 MHz clock in half, twice. This corresponds to a refresh every 64 cycles of the 4 MHz clock, or approximately 400 cycles of the 25 MHz clock. The type refresh used is $\overline{CAS}$ before $\overline{RAS}$ ($\overline{CAS}$ signal asserted before the $\overline{RAS}$ signal) for specified lengths of time. Both banks of DRAM are refreshed at the same time.

1.5 Burst Capabilities for a 32-bit Bus

The i960 Jx processor can access up to four data words per request. A request starts when the processor asserts $\overline{ADS}$ in the address cycle and ends when the processor asserts $\overline{BLAST}$ in the last data cycle. Figure 1 shows $\overline{ADS}$ and $\overline{BLAST}$ timings for a quad-word write request.

Figure 1. Quad-Word Write Request with 2,1,1,1 Wait State Profile

The processor's burst capabilities on a 32-bit bus include:

- Quad-word and triple-word requests start on quad-word boundaries (A3 = 0, A2 = 0).
- Double-word requests start on double-word boundaries (A3 = X, A2 = 0).
- Single-word requests can start on any word boundary (A3 = X, A2 = X).
- Any request starting on an odd word boundary never bursts (A3 = X, A2 = 1).

1.6 Bank Interleaving

Bank interleaving allows the second, third and fourth accesses of a burst read to occur in zero wait states. The first data access must still "pay" the entire access penalty. Interleaving significantly improves memory system performance by overlapping accesses to consecutive addresses. Two-way interleaving is accomplished by dividing the memory into two 32-bit banks (also referred to as "leaves"):

- one bank for even word addresses (A2=0)
- one bank for odd word addresses (A2=1)

The two banks are read in parallel and the data from the two banks is multiplexed into the processor's data bus. Multiplexing is implemented via $\overline{CASAx}$ and $\overline{CASBx}$. $\overline{CAS}$ signals, in addition to being the Column Address Strobes, are also output enable signals for the DRAM.

Figure 2 shows DRAM with a 2-1-1-1 quad word burst read wait state profile interleaved to generate a 2-0-0-0 wait state system.

A = Address W = Wait D = Data

Figure 2. Two-Way Interleaving

1.7 i960 Jx Processor Address Mapping

Figure 3 portrays the memory map for the i960 Jx processor to SAR interface. The DRAM beginning location is set at A0000000H and the SAR beginning location is set at C0000000H. When configuring the processor's memory regions using the PMCON registers, make sure regions PMCON10_11 and PMCON12_13 are set for 32 bit wide bus accesses.

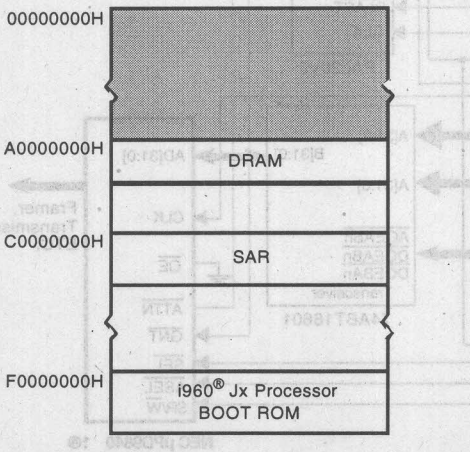


Figure 3. 80960Jx Interface Memory Map

AV9155-01 feeds the 25 MHz clock into a CY7BB991-7 with an internal PLL. Also, the AV9155-01 directly generates the 16 MHz clock that is later divided down to 4 MHz, used for DRAM refresh generation.

Clock distribution is performed by the CY7BB991-7. Seperate output clocks from this device are distributed back to the CPU module, Squall II Module, and on-board logic. This device guarantees a maximum skew of ±250 ps between outputs, and ±500 ps between the inputs and outputs. Therefore, all clocks on the board are within ±1 ns, making the design work very straightforward. All clock signals are terminated with 22 ohm series resistors.

2.2 System Address Latches

The i960 Jx processor has a multiplexed bus; therefore, address latches are needed to demultiplex the bus for use in accessing the DRAM. The latches, shown in Figure 5, capture the upper 28 address bits of the bus during the processor's address cycle, then holds them until the access is over. ALE provides the signal for latching, and HOLDA controls the output enables of the latches. HOLDA isolates the address latches from the system bus when the SAR uses the bus as a master device. As a master device, the SAR requests the bus with HOLD; the processor grants the bus with HOLDA. When HOLDA is high, the outputs to the latches are three-stated.

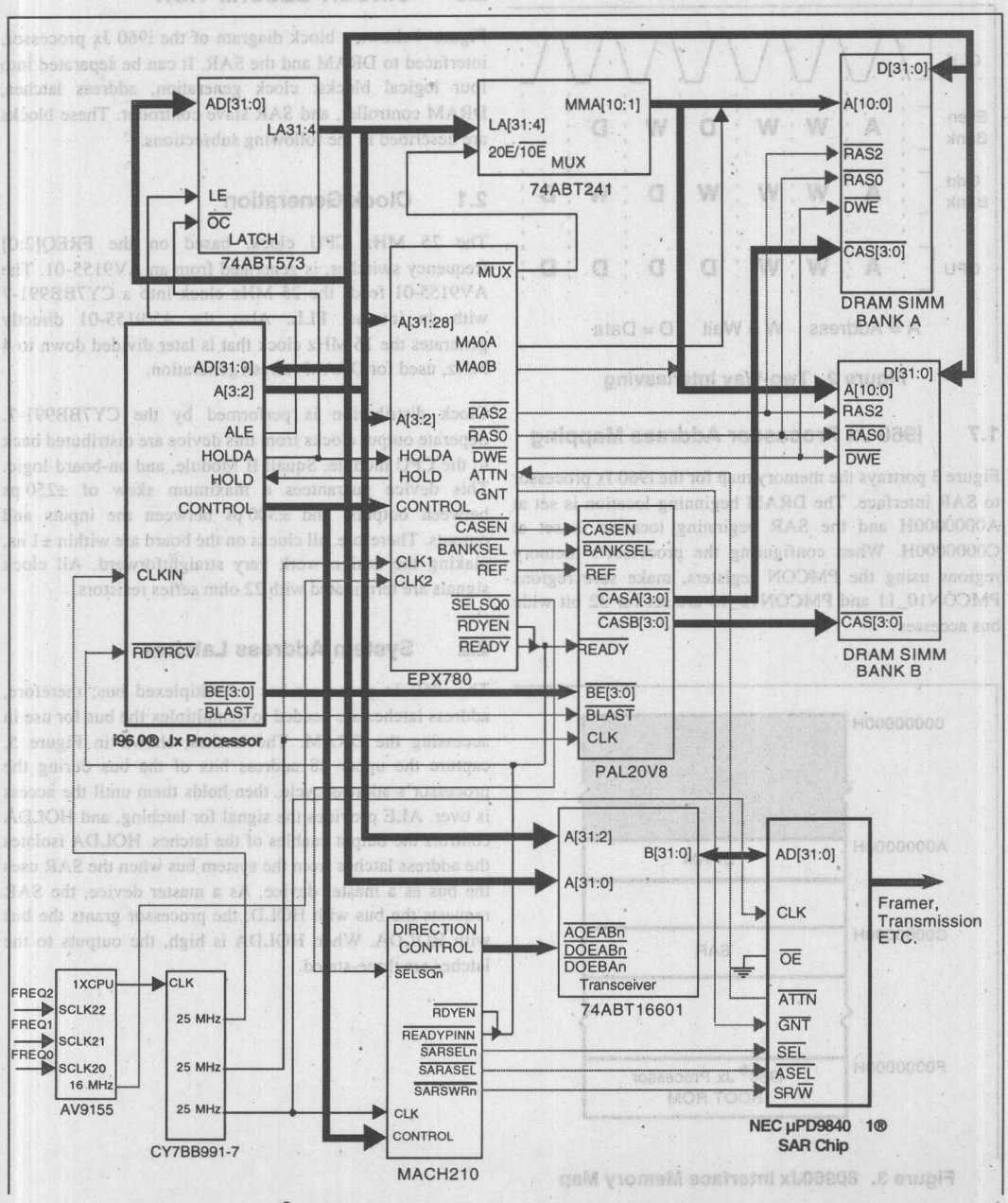


Figure 4. i960® Jx Processor to DRAM / NEC µPD98401 SAR Bus Interface

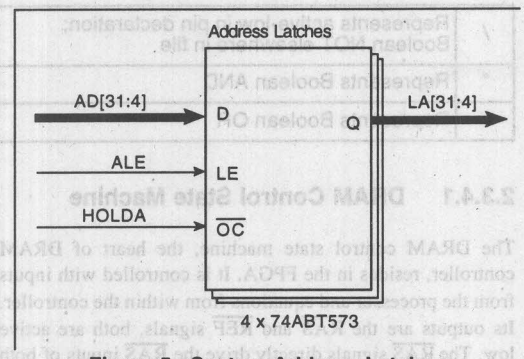


Figure 5. System Address Latches

2.3 DRAM Controller

The DRAM controller can be separated into four sections: control logic, address flow logic, data flow logic, and DRAM controller state machines and signals. See Figure 4.

2.3.1 Control Logic

The control logic is implemented using an EPX780 FPGA and a 20V8 PAL. The FPGA, the main component of the DRAM controller, supplies:

- $\overline{RAS}$ lines and read/write signals directly to the DRAM
- address bits to each DRAM bank that increments the column address
- logic to control the address path to the DRAM

The 20V8 PAL, with direction from the FPGA, provides all the necessary logic to generate the $\overline{CAS}$ signals for the DRAM.

2.3.2 Address Flow Logic

Figure 6 shows the DRAM address flow logic. The address flow logic to the DRAM is controlled by three 74ABT241 octal buffers, configured as a 12-input, two-line to one-line multiplexer. The $\overline{MUX}$ select signal, which controls the multiplexing function, is supplied by the FPGA. The multiplexers divide the latched address bus into two addresses: the row address consisting of the latched address' higher order bits, and the column address consisting of the latched address' lower order bits.

Both row and column addresses are transferred to DRAM via the MMA[10:1] address bus. MA0a connects to A0 of the bank A DRAM, MA0b connects to A0 of the bank B DRAM.

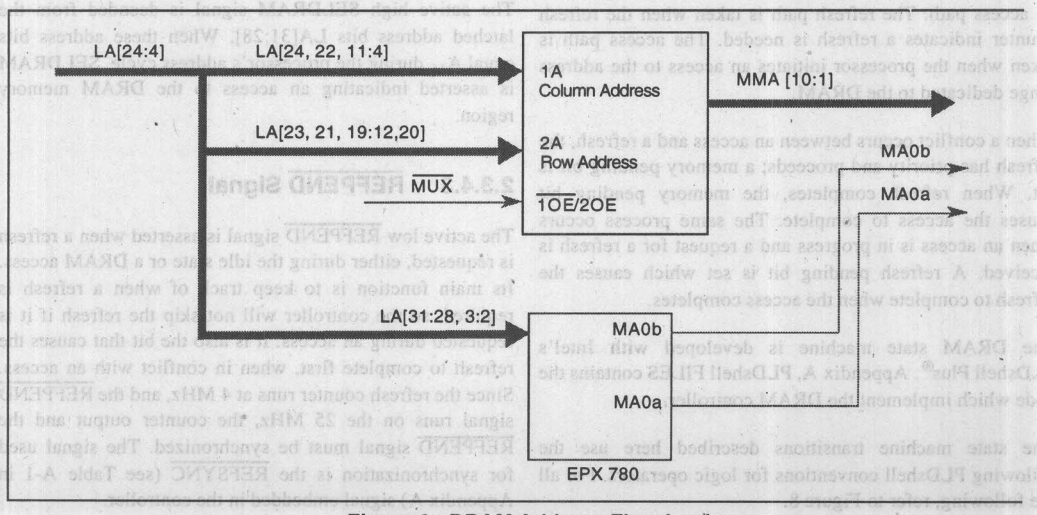


Figure 6. DRAM Address Flow Logic

2.3.4 State Machines and Signals

Figure 7 shows the DRAM control state machine for the PCI-SDK; Figure 8 shows a simplified version. The simplified version is the result of choosing the configuration discussed in Section 1.2, Overview (pg. 1-1). For this configuration, the CPU module Frequency Switches are set to 25 MHz (Positions 1-4 should be OFF, OFF, ON, ON. This maps to a logic 1,1,0,0. Refer to Cyclone documentation for definitions).

The PD3 pin output from the DRAM SIMM, when sampled high, indicates 60 ns DRAM. With these frequency and speed settings, the signal PF0 is set high and all the other PF signals are reset low. The PCI-SDK does not use the EXTEND signal; it is tied high.

The state machine has two major paths: a refresh path and an access path. The refresh path is taken when the refresh counter indicates a refresh is needed. The access path is taken when the processor initiates an access to the address range dedicated to the DRAM.

When a conflict occurs between an access and a refresh, the refresh has priority and proceeds; a memory pending bit is set. When refresh completes, the memory pending bit causes the access to complete. The same process occurs when an access is in progress and a request for a refresh is received. A refresh pending bit is set which causes the refresh to complete when the access completes.

The DRAM state machine is developed with Intel's PLDshell Plus®. Appendix A, PLDshell FILES contains the code which implement the DRAM controller.

The state machine transitions described here use the following PLDshell conventions for logic operators. For all the following, refer to Figure 8.

from the processor and equations from within the controller. Its outputs are the $\overline{RAS}$ and $\overline{REF}$ signals, both are active low. The $\overline{RAS}$ signals directly drive the $\overline{RAS}$ inputs of both banks and are used to latch the row addresses into the DRAM. Two identical $\overline{RAS}$ signals distribute the load between the two banks. $\overline{REF}$ provides a bit to the $\overline{REFPEND}$ state machine, indicating a refresh has begun. This resets the $\overline{REFPEND}$ signal, so it can be set the next time a refresh is required.

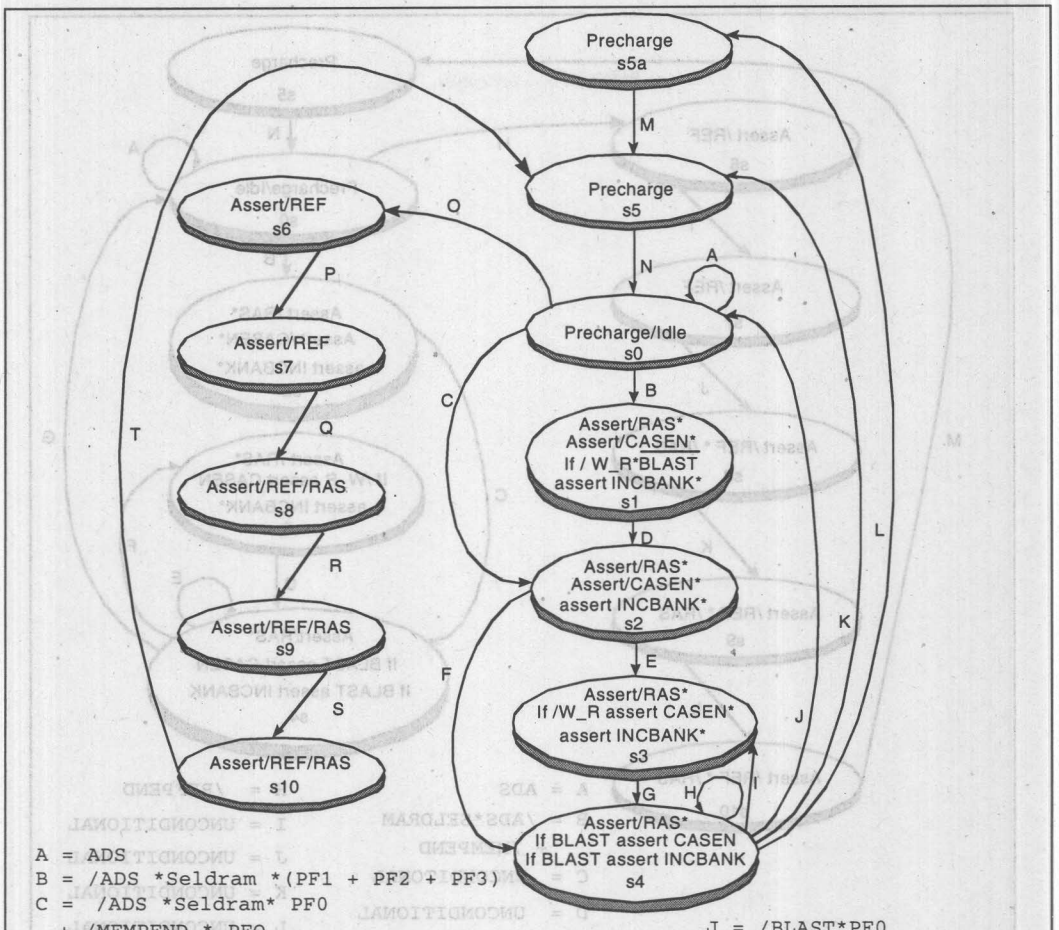
The other output functions of the DRAM controller depend upon which state the DRAM control state machine is in. The following are the signals that make up the DRAM controller.

2.3.4.2 SELDRAM Signal

The active high SELDRAM signal is decoded from the latched address bits LA[31:28]. When these address bits equal A_{16} during the processor's address cycle, SELDRAM is asserted indicating an access to the DRAM memory region.

2.3.4.3 $\overline{REFPEND}$ Signal

The active low $\overline{REFPEND}$ signal is asserted when a refresh is requested, either during the idle state or a DRAM access. Its main function is to keep track of when a refresh is required, so the controller will not skip the refresh if it is requested during an access. It is also the bit that causes the refresh to complete first, when in conflict with an access. Since the refresh counter runs at 4 MHz, and the $\overline{REFPEND}$ signal runs on the 25 MHz, the counter output and the $\overline{REFPEND}$ signal must be synchronized. The signal used for synchronization is the $\overline{REFSYNC}$ (see Table A-1 in Appendix A) signal embedded in the controller.




```
A = ADS
B = /ADS *Seldram *(PF1 + PF2 + PF3)
C =  /ADS *Seldram* PF0
  + /MEMPEND * PFO
D =  UNCONDITIONAL
E = /W_R * (PF2 + PF3)
F =  W_R * (PF2 + PF3)
  +PF1
  +PF0
G = UNCONDITIONAL
H =  BLAST*/W_R *PF0
  + BLAST*/W_R *PF1
  + /EXTEND */W_R
I = BLAST* /W_R *PF3
 + BLAST*/W_R*PF2
 + BLAST*W_R
```

```
J = /BLAST*PF0
K = /BLAST*PF3*/W_R
  + /BLAST *PF1
  + /BLAST *PF2
L = /BLAST *PF3*W_R
M = UNCONDITIONAL
N = UNCONDITIONAL
O = /REFPEND
P = UNCONDITIONAL
Q = UNCONDITIONAL
R = UNCONDITIONAL
S = UNCONDITIONAL
T = UNCONDITIONAL
```

Figure 7. DRAM Control State Machine

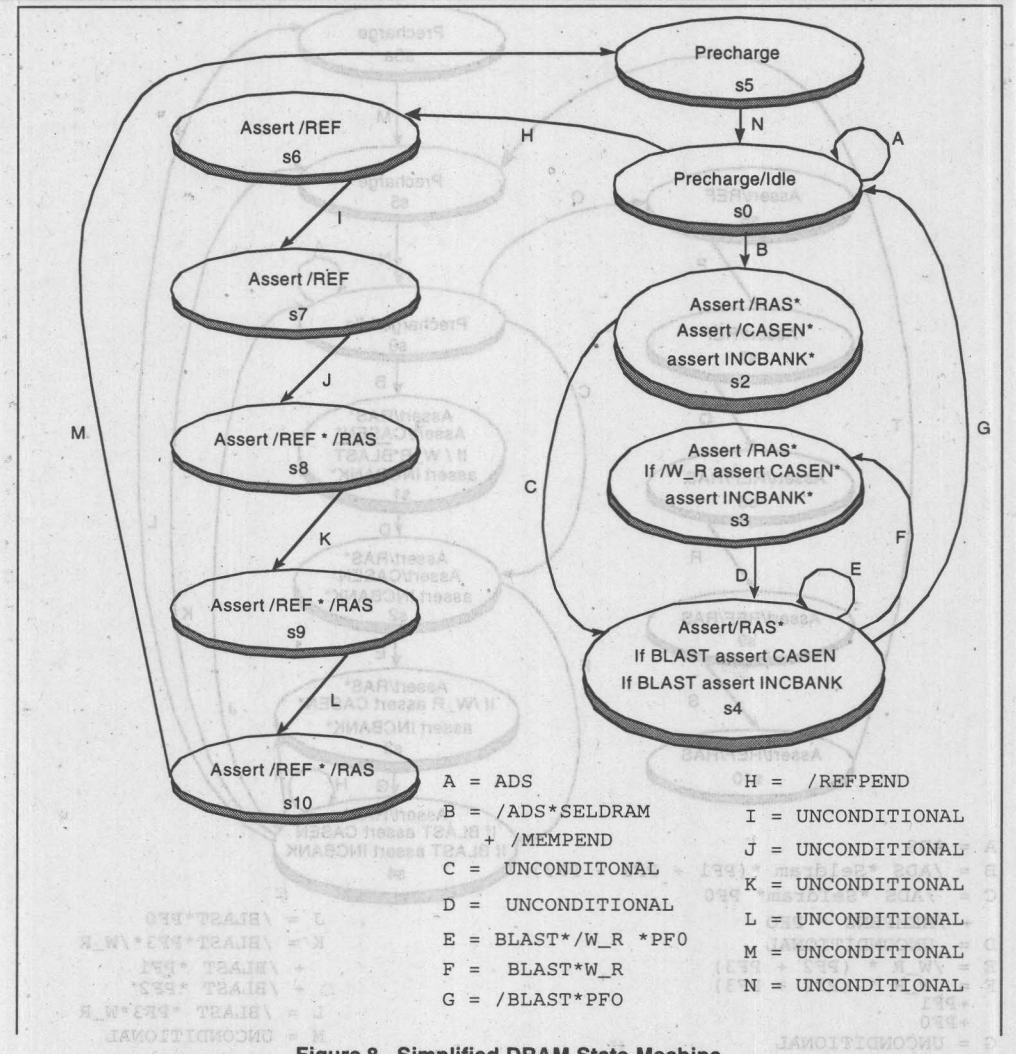


Figure 8. Simplified DRAM State Machine

2.3.4.4 $\overline{\text{MEMPEND}}$ Signal

The active low $\overline{\text{MEMPEND}}$ signal function is similar to the $\overline{\text{REFPEND}}$ signal, except it keeps track of an access when the controller is performing a refresh. It is asserted when an access is requested either during the idle state or a refresh.

2.3.4.5 $\overline{\text{MUX}}$ Signal

The $\overline{\text{MUX}}$ signal, clocked by the falling edge of the 25 MHz clock, controls whether a row or column address is input to DRAM. It is actually a $\overline{\text{RAS}}$ signal delayed by a half clock cycle. The delay allows enough row address hold time (t_{RAH}) which DRAM requires. When $\overline{\text{MUX}}$ is high, the row address to DRAM is valid; when $\overline{\text{MUX}}$ is low, the column address is valid.

2.3.4.6 $\overline{\text{CASEN}}$ Signal

The active low $\overline{\text{CASEN}}$ signal provides an output enable signal to the $\overline{\text{CAS}}$ state machines, located in the PAL20V8. Its function is determined by which state the DRAM Control state machine is in:

- s0: asserted during an address cycle in which one or more are true:
  - — access is to the DRAM memory region
  - — a refresh is not pending
  - — an access is pending
- s2 and s3: $\overline{\text{CASEN}}$ is asserted when the access is a read. (Remember that PF0 is asserted, indicating the system is operating at 25 MHz with 60 ns DRAM.)
- s4: asserted during a burst read or write access EXCEPT for the last access of the burst.

2.3.4.7 $\overline{\text{INCBANK}}$ and $\overline{\text{BANKSEL}}$ Signals

The active low $\overline{\text{BANKSEL}}$ signal determines which bank will have its $\overline{\text{CAS}}$ lines driven:

- When low, selects bank A
- When high, selects bank B

The active low $\overline{\text{INCBANK}}$ signal, in conjunction with the $\overline{\text{BANKSEL}}$ signal, creates wait states by controlling the $\overline{\text{CAS}}$ lines. When asserted low at the end of a clock cycle, the $\overline{\text{CAS}}$ lines are asserted, thereby selecting each bank. The $\overline{\text{INCBANK}}$ signal operates based upon which state the DRAM Control state machine is in.

- s0: asserted during an address cycle, in which one or more are true:
  - — access is to the DRAM memory region
  - — a refresh is not pending
  - — an access is pending
- s2: asserted if the access is a burst read, EXCEPT for the last access of the burst.
- s4: asserted during a burst read or write access, EXCEPT for the last access of the burst.

2.3.4.8 $\overline{\text{CASA[3:0]}}$ and $\overline{\text{CASB[3:0]}}$ Signals

The active low $\overline{\text{CASA}}$ and $\overline{\text{CASB}}$ signals connect to the $\overline{\text{CAS}}$ pins of DRAM banks A and B respectively. They are enabled by the $\overline{\text{CASEN}}$ and $\overline{\text{BANKSEL}}$ signals. With $\overline{\text{BANKSEL}}$ low bank A is selected, with $\overline{\text{BANKSEL}}$ high bank B is selected. These signals serve two purposes: to latch the column addresses into the DRAM, and to provide chip selects for each bank. $\overline{\text{CASA3}}$ and $\overline{\text{CASB3}}$ enables the most significant byte, and $\overline{\text{CASA0}}$ and $\overline{\text{CASB0}}$ enable least significant byte. The byte enables, $\overline{\text{BE[3:0]}}$ control which byte(s) to be transferred within each bank, allowing 8 and 16 bit wide data transfers. $\overline{\text{BE[3:0]}}$ controls bytes 3-0 respectively.

2.3.4.9 MA0B and MA0A Signals

MA0B and MA0A are the least significant column address bits that increment the address to DRAM banks B and A respectively, during a burst access. They are decoded from address bits A3 and A2 of the processor's bus, which increment during a burst.

2.3.4.10 $\overline{\text{RDYEN}}$ Signal

The active low $\overline{\text{RDYEN}}$ signal enables the three-state buffer that controls the $\overline{\text{READY}}$ signal. Since the system has two controllers on it, i.e. a DRAM controller and a SAR controller, the $\overline{\text{READY}}$ signal of the DRAM controller has to be three-stated when the SAR controller drives its $\overline{\text{READY}}$ signal. This prevents contention between the two ready signals. The $\overline{\text{RDYEN}}$ signal is asserted based upon which state the DRAM control state machine is in. It's asserted during states s0, s2, s3 and s4, regardless if $\overline{\text{READY}}$ is asserted or not.

2.3.4.11 $\overline{\text{READY}}$ Signal

The active low $\overline{\text{READY}}$ signal is connected to the $\overline{\text{RDYRCV}}$ pin of the processor. It indicates to the processor that data can be transferred, and that the address can be incremented if the access is a burst. It is asserted based upon which state the DRAM control state machine is in. It is asserted unconditionally in states s2 and s3. It is asserted in s4 if the access is a burst read, provided it's not the last access of the burst.

2.3.4.12 $\overline{\text{DWE}}$ Signal

The active low $\overline{\text{DWE}}$ signal indicates to the DRAM that the access is a write, provided the controller is not in refresh. The write enable signal of the DRAM's must be held high during refresh.

2.4 SAR slave controller

The SAR slave controller controls access to the SAR in slave mode. Reading and writing the SAR in slave mode accesses 32 bit registers that initialize and control the SAR. Referring to Figure 4, the SAR slave controller is discussed in 4 sections: control logic, address flow logic, data flow logic, and the SAR slave controller state machines and signals.

2.4.1 Control Logic

The control logic is implemented using the EPX780 FPGA and a MACH210 PLD. The FPGA decodes the SAR's address space, and provides the select signal to the SAR's access state machine. The MACH210 PLD provides all the logic that makes up the SAR controller, which controls the sequencing of the SAR itself, and the signals that control the SAR's address and data transceivers.

2.4.2 Address Flow Logic

Figure 9 shows the SAR Address Buffers. The address for the SAR is controlled by two 74ABT16601 universal bus transceivers. The $\overline{\text{AOEABn}}$ control signal, output from the SAR controller, enables the latched address bits onto the multiplexed AD bus of the SAR. When inactive, the outputs of the transceivers are three-stated, allowing data to be driven on the data bus without being in contention with the address bus.

2.4.3 Data Flow Logic

Figure 9 shows the SAR Data Buffers. The data for the SAR is controlled by two 74ABT16601 universal bus transceivers. The $\overline{\text{DOEABn}}$ control signal, output from the SAR controller, enables the data onto the multiplexed AD bus of the SAR, when the SAR is being written to. The $\overline{\text{DOEBAn}}$ control signal, output from the SAR controller, enables the data onto the AD bus of the processor, when the SAR is being read. When either control signal is inactive, both sides A and B of the data buffers are three-stated, allowing the bus to be used for other functions.

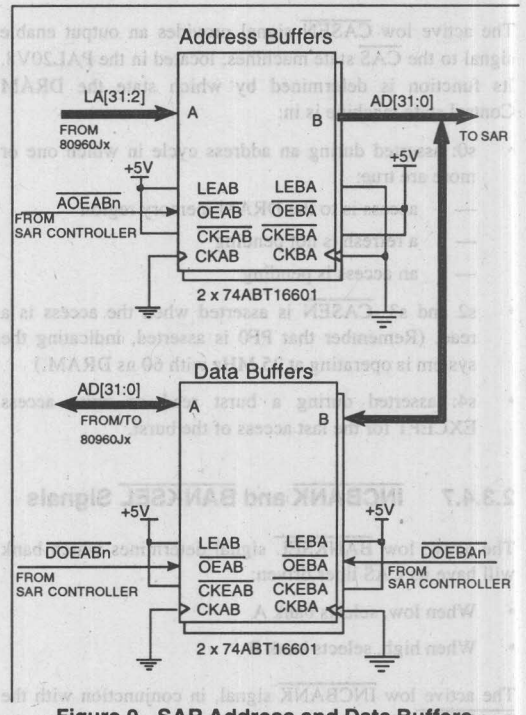


Figure 9. SAR Address and Data Buffers

2.4.4 State Machine and Signals

The SAR slave controller consists of the SAR access state machine, and several equations that implement the control signals. The SAR controller resides in the MACH210 PLD.

The SAR controller is written in MACHXL. Appendix B, PLD EQUATIONS contain the code which implement the SAR controller. The state machine transitions described here follow the MACHXL conventions for logic operators.

/	Represents Boolean NOT
*	Represents Boolean AND
+	Represents Boolean OR
{}	Represents Substitution (OUT1=A*B*C, OUT2=A*B*C*D or OUT2={OUT1}*D)

2.4.4.1 SELC Signal

The active high SELC signal is decoded from the latched address bits LA[31:28]. When these address bits equal C_{16} during the processor's address cycle, SELC is asserted indicating an access to the SAR memory region.

2.4.4.2 SELSQ0 Signal

$\overline{\text{SELSQ0}}$ is the select signal (standing for Select Squall module, remembering that the SAR is on the squall module) which initiates the SAR Access State Machine when SELC is asserted.

2.4.4.3 SAR Access State Machine

Shown in Figure 10 is the SAR access state machine. It is initiated when the $\overline{\text{SELSQ0}}$ select signal from the FPGA is asserted, due to a request issued to the SAR's memory region.

The state of the control signals implemented in the controller, depend upon which state the state machine is in. The following are the signals that make up the SAR controller.

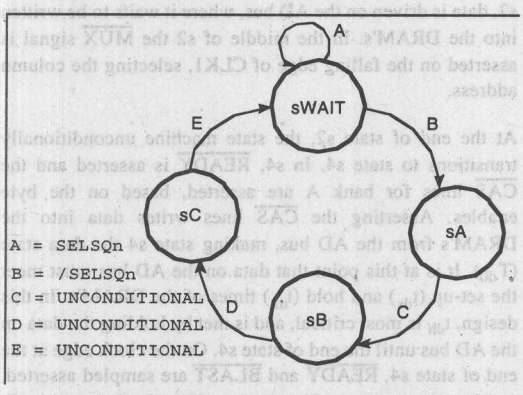


Figure 10. 80960Jx to SAR Access State Machine

2.4.4.4 $\overline{\text{READYn}}$ Signal

The $\overline{\text{READYn}}$ signal is asserted on the rising edge of the clock, at the end of state sB of the SAR access state machine. At the end of state sC, $\overline{\text{READYn}}$ signals to the processor the termination of a data transfer.

2.4.4.5 $\overline{\text{READYpinn}}$ Signal

The $\overline{\text{READYpinn}}$ signal is connected to the $\overline{\text{RDYRCV}}$ pin of the processor. It can be three-stated and is driven by $\overline{\text{READYn}}$. It must be three-stated when an access other than a SAR access occurs. The $\overline{\text{READYpinn}}$ signal is paired with the $\overline{\text{READYn}}$ signal, indicated by the brackets around $\overline{\text{READYn}}$ on the right side of the $\overline{\text{READYpinn}}$ equation. (see the MACHXL PLD file of Appendix B). Pairing connects a node signal like $\overline{\text{READYn}}$, to an output signal like $\overline{\text{READYpinn}}$.

2.4.4.6 $\overline{\text{RDYENn}}$ Signal

$\overline{\text{RDYENn}}$ controls the three-state buffer that three-states the $\overline{\text{READYpinn}}$ signal. It is asserted when $\overline{\text{SELSQ0}}$ is asserted.

2.4.4.7 $\overline{\text{SARSELn}}$ Signal

The $\overline{\text{SARSELn}}$ signal selects the SAR, enabling it to be accessed in slave mode.

2.4.4.8 $\overline{\text{SARASELn}}$ Signal

The $\overline{\text{SARASELn}}$ signal indicates to the SAR that an address is on its AD bus. On the first rising clock edge after $\overline{\text{SARASELn}}$ goes active, the SAR latches the address onto its AD bus for use during an access.

2.4.4.9 $\overline{\text{SARSWRn}}$ Signal

The $\overline{\text{SARSWRn}}$ signal determines the direction of the slave access. 1=read, 0=write.

2.4.4.10 $\overline{\text{AOEABn}}$ Signal

The $\overline{\text{AOEABn}}$ signal enables the B output of the SAR's address buffers, directing the flow of the latched address from the PCI-SDK, onto the AD bus of the SAR.

2.4.4.11 $\overline{\text{DOEABn}}$ Signal

The $\overline{\text{DOEABn}}$ signal enables the flow of data through the data buffers from A to B, to the AD bus of the SAR during a write.

2.4.4.12 $\overline{\text{DOEBAn}}$ Signal

The $\overline{\text{DOEBAn}}$ signal enables the flow of data through the data buffers from B to A, to the AD bus of the processor during a read.

3.0 DRAM CONTROLLER ACCESS FLOW

This section explains how the DRAM controller is sequenced while reading, writing and refreshing the DRAM. Examples discussed are:

- single word read access
- quad word read access
- single word write access
- quad word write access
- refresh

For the following accesses, refer to Figure 8. All accesses to the DRAM are with the i960 Jx processor configured with a 32 bit bus.

3.1 Single Word Read and Write Access

Figure 11 shows a single word read timing diagram. In state s0 of the read when an access has begun, the address bits from the processor are latched, and held latched throughout the access. The $\overline{\text{MUX}}$ signal stays deasserted, selecting the row address via the multiplexer. On the clock edge at the end of state s0, the DRAM control state machine transitions to state s2. In the beginning of state s2, the $\overline{\text{BLAST}}$ signal, $\overline{\text{MEMPEND}}$ bit, $\overline{\text{INCBANK}}$ signal, and all the $\overline{\text{RAS}}$ lines are asserted. $\overline{\text{BLAST}}$ asserting at this time indicates a single access, $\overline{\text{MEMPEND}}$ indicates an access is pending, $\overline{\text{INCBANK}}$ indicates the next state is a data state, while the $\overline{\text{RAS}}$ lines load the row address into all the DRAM's. In the middle of s2 the $\overline{\text{MUX}}$ signal is asserted on the falling edge of CLK1, selecting the column address.

At the end of state s2, the state machine unconditionally transitions to state s4. In s4, $\overline{\text{READY}}$ is asserted and the $\overline{\text{CAS}}$ lines for bank A are asserted, based on the byte enables. Asserting of the $\overline{\text{CAS}}$ lines enables data from the DRAM's onto the AD bus, making state s4 the data state (T_{d0}). On the clock edge at the end of state s4, data is captured by the processor due to sampling $\overline{\text{READY}}$ asserted. It is at this point that data put on the AD bus from the DRAM, must meet the set-up (t_{is1}) and hold (t_{ih1}) times of the processor. At the time $\overline{\text{READY}}$ is sampled low, $\overline{\text{BLAST}}$ is sampled low, causing the state machine to transition to state s0, completing the access and deasserting the $\overline{\text{MEMPEND}}$ bit.

Figure 11 shows a single word write timing diagram. This access sequences in the same manner as the single word read access did, up until the first data state.

In state s0 of the write when an access has begun, the address bits from the processor are latched, and held latched throughout the access. The $\overline{\text{MUX}}$ signal stays deasserted, selecting the row address via the multiplexer. On the clock edge at the end of state s0, the DRAM control state machine transitions to state s2, due to the processor access. In the beginning of state s2, the $\overline{\text{BLAST}}$ signal, $\overline{\text{MEMPEND}}$ bit, $\overline{\text{INCBANK}}$ signal, and all the $\overline{\text{RAS}}$ lines are asserted. $\overline{\text{BLAST}}$ asserting at this time indicates a single access, $\overline{\text{MEMPEND}}$ indicates an access is pending, $\overline{\text{INCBANK}}$ indicates the next state is a data state, while the $\overline{\text{RAS}}$ lines load the row address into all the DRAM's. Also during state s2, data is driven on the AD bus, where it waits to be written into the DRAM's. In the middle of s2 the $\overline{\text{MUX}}$ signal is asserted on the falling edge of CLK1, selecting the column address.

At the end of state s2, the state machine unconditionally transitions to state s4. In s4, $\overline{\text{READY}}$ is asserted and the $\overline{\text{CAS}}$ lines for bank A are asserted, based on the byte enables. Asserting the $\overline{\text{CAS}}$ lines writes data into the DRAM's from the AD bus, making state s4 the data state (T_{d0}). It is at this point that data on the AD bus must meet the set-up (t_{ds}) and hold (t_{dh}) times of the DRAM's. In this design, t_{dh} is most critical, and is met by holding the data on the AD bus until the end of state s4. On the clock edge at the end of state s4, $\overline{\text{READY}}$ and $\overline{\text{BLAST}}$ are sampled asserted, causing the state machine to transition back to state s0, completing the access and deasserting the $\overline{\text{MEMPEND}}$ bit.

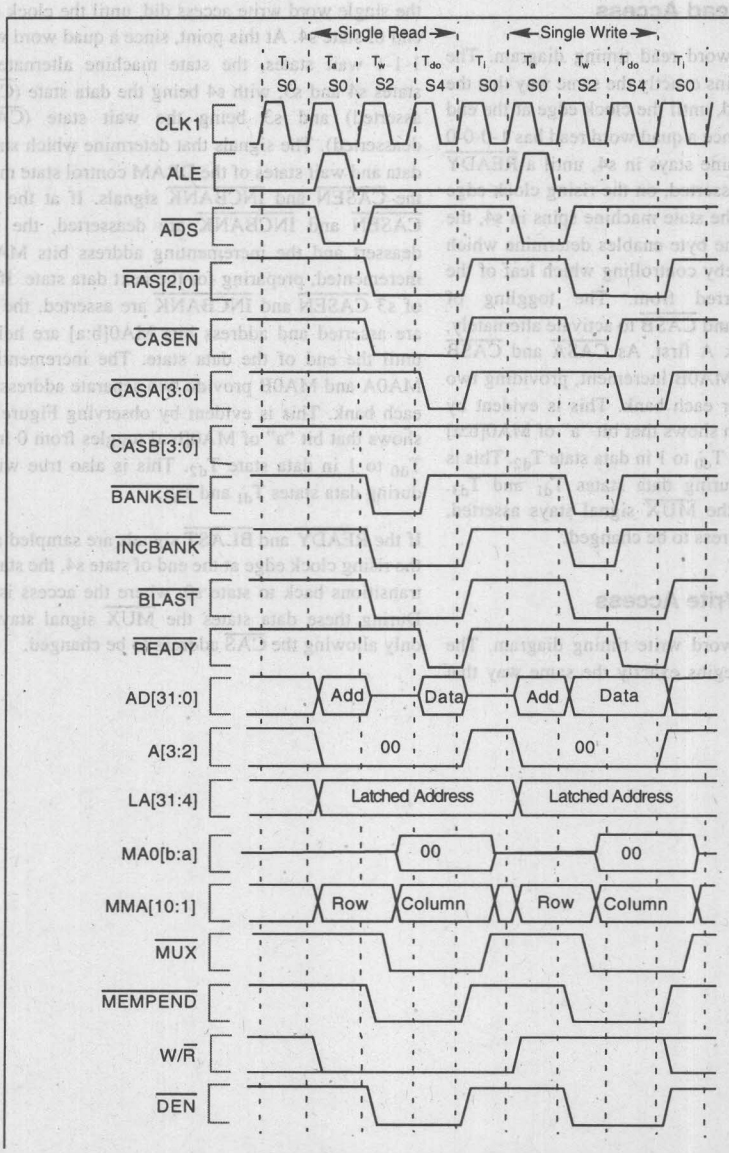
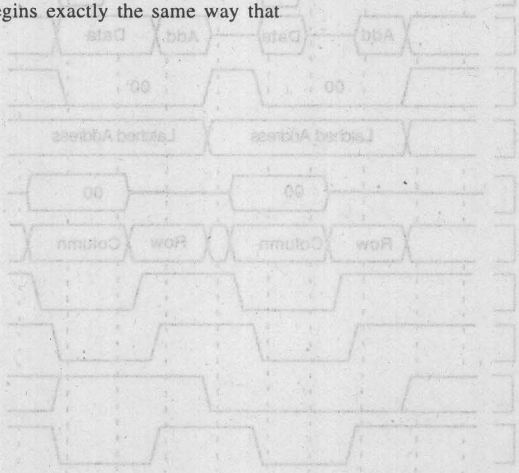


Figure 11. 80960Jx to DRAM 1 Wait State Single Word Read and Write Timing Diagram

CAS signals activate, thereby controlling which leaf of the DRAM data is transferred from. The toggling of $\overline{\text{BANKSEL}}$ causes $\overline{\text{CASA}}$ and $\overline{\text{CASB}}$ to activate alternately, with $\overline{\text{CASA}}$ enabling bank A first. As $\overline{\text{CASA}}$ and $\overline{\text{CASB}}$ alternate, bits MA0A and MA0B increment, providing two separate addresses, two for each bank. This is evident by observing Figure 12, which shows that bit "a" of MA0[b:a] toggles from 0 in data state T_{d0} to 1 in data state T_{d2}. This is also true with bit "b", during data states T_{d1} and T_{d3}. During these data states the $\overline{\text{MUX}}$ signal stays asserted, only allowing the $\overline{\text{CAS}}$ address to be changed.

3.3 Quad Word Write Access

Figure 13 shows a quad word write timing diagram. The quad word write access begins exactly the same way that



incremented, preparing for the next data state. If at the end of s3 $\overline{\text{CASEN}}$ and $\overline{\text{INCBANK}}$ are asserted, the $\overline{\text{CAS}}$ lines are asserted and address bits MA0[b:a] are held constant until the end of the data state. The incrementing of bits MA0A and MA0B provide two separate addresses, two for each bank. This is evident by observing Figure 13, which shows that bit "a" of MA0[b:a] toggles from 0 in data state T_{d0} to 1 in data state T_{d2}. This is also true with bit "b", during data states T_{d1} and T_{d3}.

If the $\overline{\text{READY}}$ and $\overline{\text{BLAST}}$ signals are sampled asserted on the rising clock edge at the end of state s4, the state machine transitions back to state s0, where the access is complete. During these data states the $\overline{\text{MUX}}$ signal stays asserted, only allowing the $\overline{\text{CAS}}$ address to be changed.

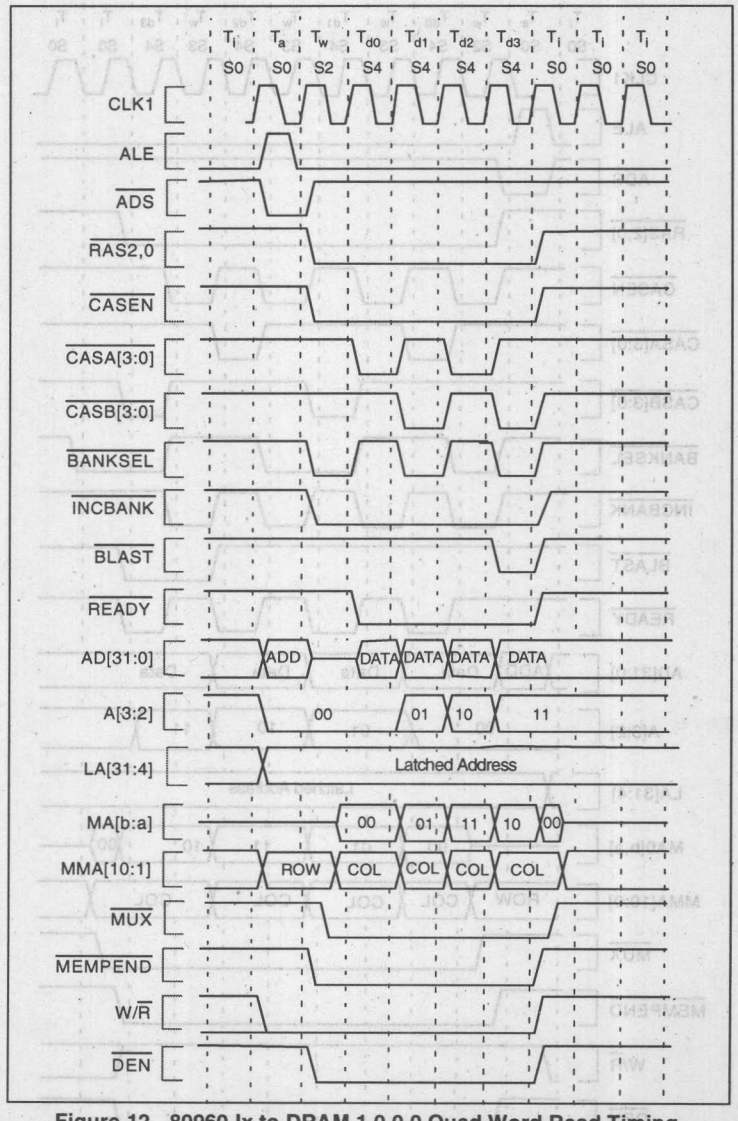


Figure 12. 80960Jx to DRAM 1,0,0,0 Quad Word Read Timing

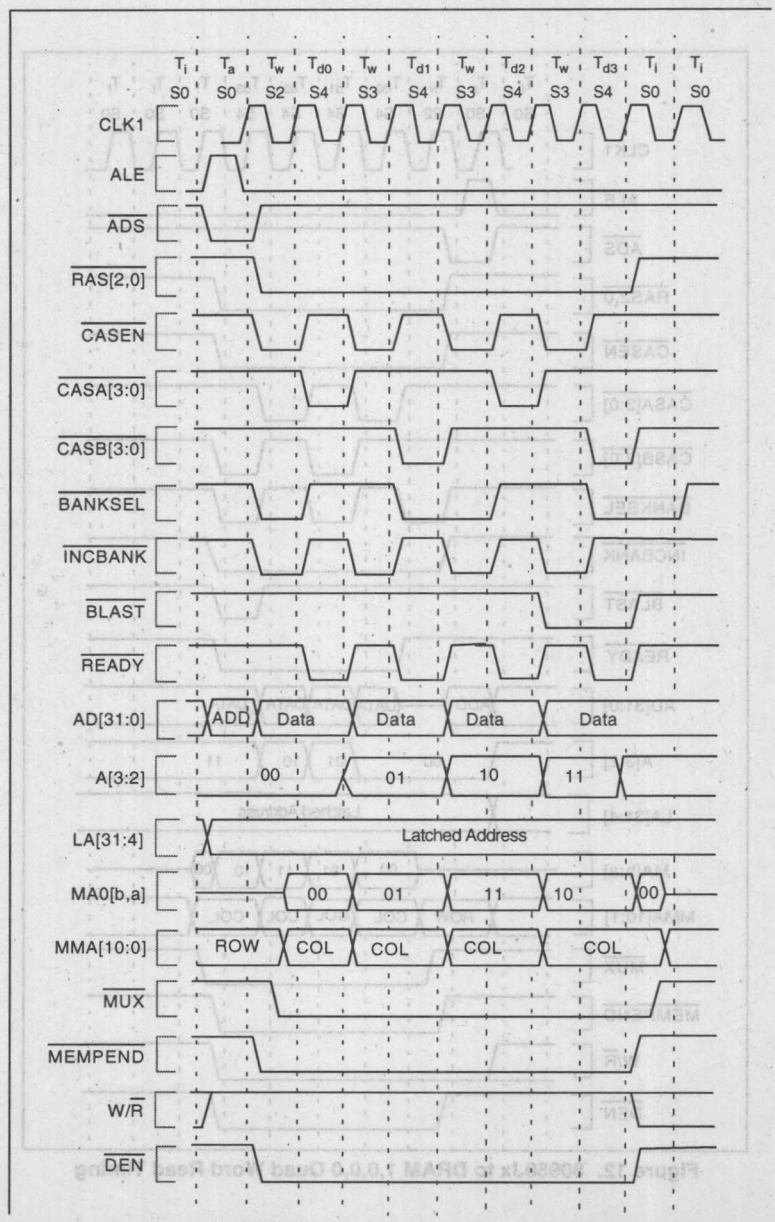


Figure 13. 80960Jx to DRAM 1,1,1,1 Quad Word Write Timing Diagram

3.4 CAS before RAS Refresh

Figure 14 shows a refresh cycle timing diagram. As soon as the PCI-SDK powers up, the refresh counter starts counting, and cannot be stopped by RESET. The counter is a free-running divide-by-64 counter, running on a 4 MHz clock. Its output switches high every 64 cycles of the 4 MHz clock, giving a 16 µs interval for requesting refresh.

On the next 25 MHz clock edge after the output of the counter switches high, the REFSYNC signal is asserted, causing the REFCLK and REFCLKD signals to sequence as shown in Figure 14. When REFCLK is sampled high and REFCLKD is sampled low on the clock edge, REFPEND is asserted. When REFPEND is detected low on the next clock edge, the DRAM control state machine enters state s6 where refresh begins. At the beginning of s6, REF is sserted and stays asserted through state s10. On the clock edge at the end of s6, REF indicates to the REFPEND state machine that the DRAM refresh is going to be satisfied, causing REFPEND to be deasserted.

The REF signal serves another purpose. It enables the CAS state machines during refresh, allowing the CAS signals to be driven. In states s8-s10, the RAS signals are asserted, giving a CAS before RAS refresh, since the CAS signals were asserted back in state s6.

After the state machine transitions out of state s10 completing the refresh, it passes through states s5 and s0, where the required RAS precharge takes place before another access can begin

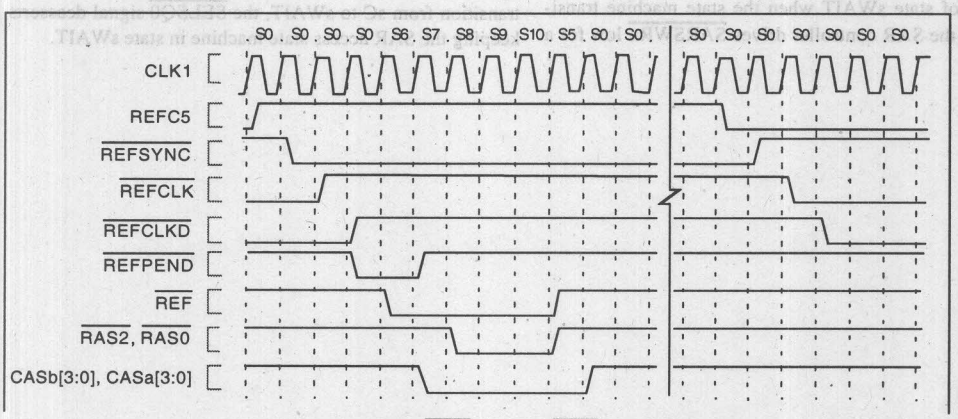


Figure 14. CAS-Before-RAS Refresh Cycle

4.0 SAR SLAVE ACCESS FLOW

This section describes how the SAR controller is sequenced during a single word read and write access.

All accesses to the SAR are with the i960 Jx processor configured with a 32 bit bus.

4.1 Single Word Read and Write Access

Figure 15 shows the Single Word Read Access Timing Diagram. The single word access begins with the processor making a request to the SAR address space. The address is latched and decoded, asserting the SELSQ0 select signal. On the first rising clock edge that SELSQ0 is low, the SAR access state machine initiates and unconditionally transitions from state sA through state sC, back into state sWAIT. The state machine will stay in sWAIT until SELSQ0 asserts, indicating another access.

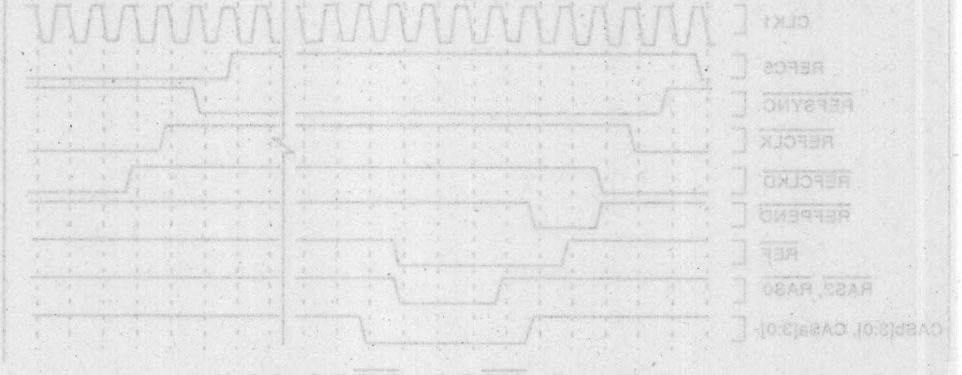
At the end of state sWAIT when the state machine transitions to sA, the SAR controller drives SARSWRn high for a read, and asserts SARSELn, SARASELn, and AOEABn. As AOEABn asserts, the address buffers place the address on the SAR's AD bus, where it is latched into the SAR on the first rising edge of the clock after SARASELn goes low. On the following clock edge after the address is latched, the SAR places the data on its AD lines, which is at the beginning of state sC, provided that SARSELn is low. At the beginning of state sC with the data placed on the SAR's AD bus, READYpinn is asserted and DOEBAn activates, placing the data onto the processors AD bus, through the

data buffers. At the end of state sC, the processor samples READY and BLAST asserted, signaling the end of the data transfer. When READY and BLAST are sampled asserted during the transition from sC to sWAIT, the SELSQ0 signal deasserts, keeping the SAR access state machine in state sWAIT.

Figure 15 shows the Single Word Write Access Timing Diagram. The single word access begins with the processor making a request to the SAR address space. The address is latched and decoded, asserting the SELSQ0 select signal. On the first rising clock edge that SELSQ0 is low, the SAR access state machine initiates and unconditionally transitions from state sA through state sC, back into state sWAIT. The state machine will stay in sWAIT until SELSQ0 asserts, indicating another access.

At the end of state sWAIT when the state machine transitions to sA, the SAR controller drives SARSWRn low for a write, and asserts SARSELn, SARASELn, and AOEABn. As AOEABn asserts, the address buffers place the address on the SAR's AD bus, where it is latched into the SAR on the first rising edge of the clock after SARASELn goes low. On the same clock edge that the address is latched, the DOEABn signal is asserted, enabling data onto the SAR's AD bus, through the data buffers. The SAR then completes the access by latching the data present on its AD bus on the leading edge of the clock, just before SARSELn goes high. The clock edge in this case is at the end of state sB.

In the beginning of state sC after the access has completed, the controller asserts READYpinn and deasserts DOEABn, removing the data from the SAR's AD bus. At the end of state sC, the processor samples READY and BLAST asserted, signaling the end of the data transfer. When READY and BLAST are sampled asserted during the transition from sC to sWAIT, the SELSQ0 signal deasserts, keeping the SAR access state machine in state sWAIT.



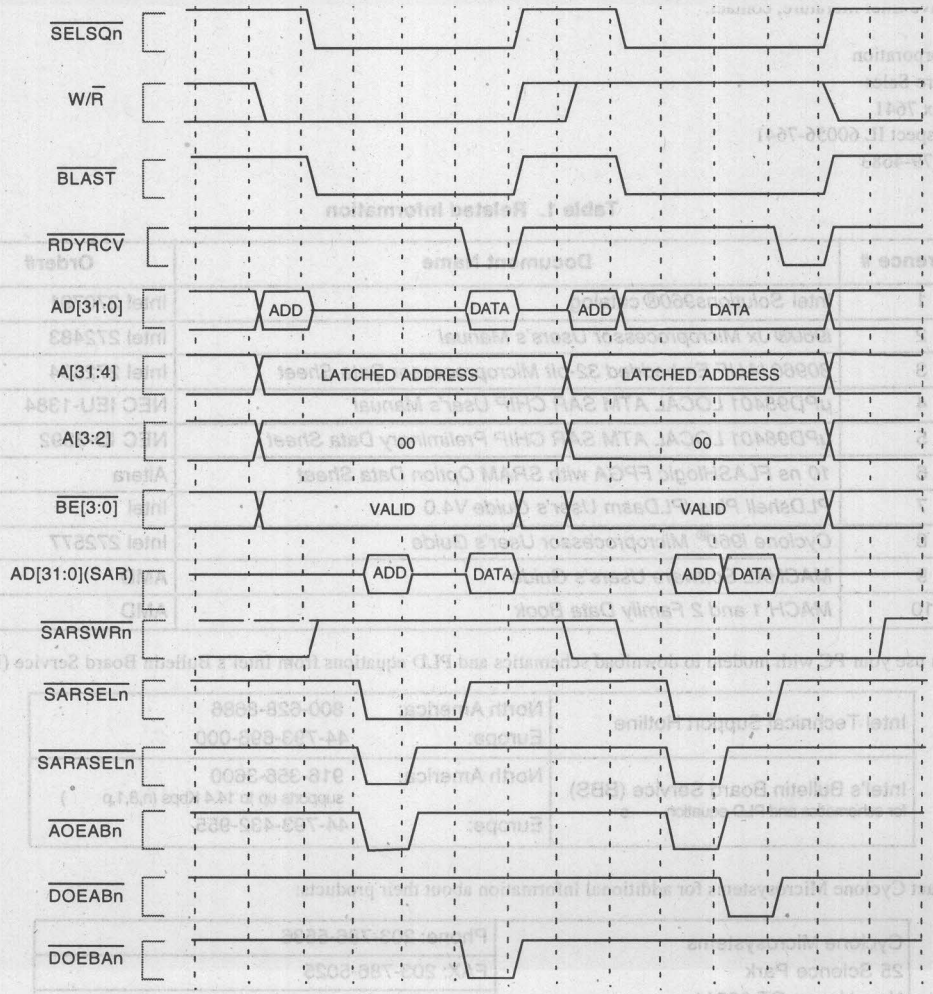


Figure 15. 80960Jx to µPD98401 Single Word Read and Write Timing Diagram

5.0 CONCLUSION

This application note discusses how to interface between an Intel i960 Jx processor and an NEC µPD98401 Local ATM SAR Chip. The platform used was a Cyclone Microsystems PCI-SDK evaluation platform, with a i960 Jx processor module and an ATM Squall module. With the Squall module containing the SAR chip. To support the interface between the i960 Jx processor and the SAR chip are 2, 8, or 32 Mbytes of DRAM, implemented in two 72 pin SIMM sockets. The schematics were generated in Future Net. The PLD equations for the DRAM controller were written in PLDshell, and the SAR controller equations were written in MACHXL. The schematics and PLD files are available through Intel America's Application Support BBS.

6.0 RELATED INFORMATION

To receive Intel literature, contact:

Intel Corporation
Literature Sales
P.O. Box 7641
Mt. Prospect IL 60056-7641
1-800-879-4683

Table 1. Related Information

Reference #	Document Name	Order#
1	Intel *Solutions960®* catalog	Intel 270791
2	*i960® Jx Microprocessor Users's Manual*	Intel 272483
3	*80960JA/JF Embedded 32-bit Microprocessor Data Sheet*	Intel 272504
4	*µPD98401 LOCAL ATM SAR CHIP User's Manual*	NEC IEU-1384
5	*µPD98401 LOCAL ATM SAR CHIP Preliminary Data Sheet*	NEC ID-3392
6	*10 ns FLASHlogic FPGA with SRAM Option Data Sheet*	Altera
7	*PLDshell Plus /PLDasm User's Guide* V4.0	Intel
8	*Cyclone i960® Microprocessor User's Guide*	Intel 272577
9	*MACHXL Software Users's Guide*	AMD
10	*MACH 1 and 2 Family Data Book*	AMD

You can use your PC with modem to download schematics and PLD equations from Intel's Bulletin Board Service (BBS).

Intel Technical Support Hotline	North America: Europe:	800-628-8686 44-793-696-000
Intel's Bulletin Board Service (BBS) for schematics and PLD equation s	North America: Europe:	916-356-3600 supports up to 14.4 Kbps (n,8,1,p) 44-793-432-955

To contact Cyclone Microsystems for additional information about their products:

Cyclone Microsystems 25 Science Park New Haven CT 06511	Phone: 203-786-5536 FAX: 203-786-5025 e-mail: info@cyclone.com

APPENDIX A
PLDshell FILES

Table A-1 contains the PLD equations used to build the portion of the DRAM controller, implemented in the Altera FLASHlogic EPX780 FPGA. Table A-2 contains the PLD equations used to build the portion of the DRAM controller, implemented in the PAL20V8. The PLD equations were created in PLDshell.

Table A-1. DRAM Controller EPX780 PLDshell File (Sheet 1 of 12)

```
TITLE          Private DRAM Control
PATTERN        D601 - Compiled with PLDShell
REVISION       E
AUTHOR         J. Smith
COMPANY        Cyclone Microsystems Inc.
DATE           5-10-95

CHIP      DCTRL          iFX780_84


PIN      3   pclk          ;INP: system clock
PIN     45   clk16         ;INP: 16MHz clock
PIN     51   rst           ;INP: reset
PIN     47   ads           ;INP:
PIN     81   blast         ;INP:
PIN     55   a31           ;INP:
PIN     56   a30           ;INP:
PIN     58   a29           ;INP:
PIN     60   a28           ;INP:
PIN     62   a19           ;INP:
PIN     64   a18           ;INP:
PIN     32   a3            ;INP: address to increment
PIN     13   a2            ;INP: select bank cas
PIN     20   a1            ;INP: select bank cas
PIN     75   w_r           ;INP: shared w/r
PIN     76   holda         ;INP:
PIN     77   sqbr0         ;INP: Squall 0 bus request
PIN     78   pcihold       ;INP: PCI 9060 bus request
PIN     79   lock          ;INP: 960 lock
PIN     82   extend        ;INP: squall module extend dram reads
PIN      1   pt2           ;INP: i960 processor type Pulled hi on board
PIN     35   pt1           ;INP: i960 processor type
PIN     37   pt0           ;INP: i960 processor type
PIN      8   freq2          ;INP: i960 processor freq
PIN      7   freq1          ;INP: i960 processor freq
PIN     28   freq0          ;INP: i960 processor freq
PIN     74   pd3           ;INP: SIMM PD3 output 70/60ns
PIN     73   swaprom        ;INP:
PIN     21   pciinstall     ;INP: PCI9060 installed = 1
PIN   IO29   readyin        ;INP: should be given same pin #
                            ; as ready pin - this is for
                            ; pin feedback
```

Table A-1. DRAM Controller EPX780 PLDshell File (Sheet 2 of 12)

```
PIN   IO51   muxin          ;INP: should be given same pin #
                            ; as mux pin.  muxalt when operating @ 33Mhz
PIN     22   irquart        ;INP: active high uart interrupt
PIN     83   breqo          ;INP: PCI9060 indicates deadlock

PIN     84   boff      REG  ;OUT: BOFF* to Cx processors @ deadlock
PIN     30   altREADY  REG  ;OUT: ready during deadlock
PIN     54   irqddlk   REG  ;OUT: deadlock interrupt
PIN      9   ras0      REG  ;OUT:
PIN      5   ras2      REG  ;OUT:
PIN     63   mux       REG  ;OUT: mux output - freq<33 , active low
PIN     61   muxalt    REG  ;OUT: mux output - freq>=33,   "      "
PIN     27   casen     REG  ;OUT:
PIN     72   banksel   REG  ;OUT:
PIN     70   ma0a      REG  ;OUT:
PIN     69   ma0b      REG  ;OUT:
PIN     31   dwe       REG  ;OUT: DRAM write enable
PIN     12   ready     REG  ;out: i960 ready
PIN      6   ref       REG  ;out:
PIN     48   hold      REG  ;OUT: hold processor
PIN     49   sqbg0     REG  ;OUT: bus grant Squall 0
PIN     50   pcihlda    REG  ;OUT: bus grant to PLX 9060

PIN     19   clk4      REG  ;OUT: 4MHz clock
PIN     18   zrd       REG  ;OUT:
PIN     16   zwr       REG  ;OUT:
PIN     15   ior       REG  ;OUT:
PIN     14   iow       REG  ;OUT:
PIN     43   selio     REG  ;
PIN     42   selcio    REG  ;OUT:
PIN     41   seluart   REG  ;OUT:
PIN     40   selpp     REG  ;OUT: select parallel port
PIN     33   selpmrom  REG  ;OUT:
PIN     34   selfrom0  REG  ;OUT:
PIN     36   selfrom1  REG  ;OUT:
PIN     39   selsq0    REG  ;OUT: Select Squall Module
PIN     57   irquartn  COMB ;OUT: active low uart interrupt

NODE  IO46   altRDYEN  REG  ;buried: ready enable during deadlock
NODE  IO28     rdyen   REG  ;buried: ready enable
NODE  IO49     st0     REG
NODE  IO44     st1     REG
NODE  IO42     st2     REG
NODE  IO31     bss0    REG  ;buried: bank select state 0
NODE  IO05     incbank REG  ;buried:
NODE  IO07     mempend REG  ;buried:
NODE  IO52     sx           ;buried: Sx processor is accessing Mem.
NODE  IO50     pfv1         ;buried:
NODE  IO57     pfv0         ;buried:
```

Table A-1. DRAM Controller EPX780 PLDshell File (Sheet 3 of 12)

```
NODE   IO54    selmux          ;buried:
NODE   IO76    refsync   REG   ;buried:
NODE   IO06    refclk    REG   ;buried:
NODE   IO08    refclkd   REG   ;buried:
NODE   IO02    refpend   REG   ;buried: refresh pending
NODE   IO67    BOOTRD0   REG   ;buried: set at reset, clr w/ ADS
NODE   IO65    BOOTRD1   REG   ;buried: set at ADS, clr w/ blast

NODE   IO22    clk8      REG   ;buried: 8MHz clock
NODE   IO10    ios0      REG   ;buried: io control states
NODE   IO13    ios1      REG   ;buried: io control states
NODE   IO16    ios2      REG   ;buried: io control states
NODE   IO15    ios3      REG   ;buried: io control states
NODE   IO19    ios4      REG   ;buried: io control states
NODE   IO32    ciotrc    REG   ;buried:
NODE   IO34    synctrc   REG   ;buried:
NODE   IO36    count0    REG   ;buried: cio trcv states
NODE   IO38    count1    REG   ;buried: cio trcv states
NODE   IO79    refc0     REG   ;buried: refresh counter
NODE   IO78    refc1     REG   ;buried: refresh counter
NODE   IO74    refc2     REG   ;buried: refresh counter
NODE   IO72    refc3     REG   ;buried: refresh counter
NODE   IO71    refc4     REG   ;buried: refresh counter
NODE   IO70    refc5     REG   ;buried: refresh counter
NODE   IO12    t0        REG   ;buried: timeout states
NODE   IO14    t1        REG   ;buried: timeout states
NODE   IO55    d0        REG   ;buried: deadlock states
NODE   IO58    d1        REG   ;buried: deadlock states

STRING    sel4_7  '/a31 *  a30'                    ;regions 4-7
STRING    sel8_9  ' a31 * /a30 * /a29'             ;regions 8-9
STRING    sel0_3  '/a31 * /a30'                    ;region  0-3
STRING    seldram ' a31 * /a30 *  a29 * /a28'      ;region  A
STRING    selb    ' a31 * /a30 *  a29 *  a28'      ;region  B
STRING    selc    ' a31 *  a30 * /a29 * /a28'      ;regions C
STRING    seld    ' a31 *  a30 * /a29 *  a28'      ;regions D
STRING    sele    ' a31 *  a30 *  a29 * /a28'      ;region  E
STRING    self    ' a31 *  a30 *  a29 *  a28'      ;region  F

STRING    pf0  '/pfv1 * /pfv0'
STRING    pf1  '/pfv1 *  pfv0'
STRING    pf2  ' pfv1 * /pfv0'
STRING    pf3  ' pfv1 *  pfv0'

STRING    Cx   '/pt2 *  pt1 * /pt0'
STRING    Jx   ' pt2 * /pt1 * /pt0'
```

```
                                                    ;52221,42221
          1     1     1   0   :     1     0         ; pf3=50MHz,      70ns
                                                    ;52222,42221

      ;********** I/O Wait & ready generation ***
STATE MEALY_MACHINE
DEFAULT_BRANCH io0
io0  = ios4 *  ios3 *  ios2 *  ios1 *  ios0
io1  = ios4 *  ios3 *  ios2 *  ios1 * /ios0
io2  = ios4 *  ios3 *  ios2 * /ios1 * /ios0
io3  = ios4 *  ios3 *  ios2 * /ios1 *  ios0
io4  = ios4 *  ios3 * /ios2 * /ios1 *  ios0
io5  = ios4 *  ios3 * /ios2 * /ios1 * /ios0
io6  = ios4 *  ios3 * /ios2 *  ios1 * /ios0
io7  = ios4 *  ios3 * /ios2 *  ios1 *  ios0
io8  = ios4 * /ios3 * /ios2 *  ios1 *  ios0
io9  = ios4 * /ios3 * /ios2 * /ios1 *  ios0
io10 = ios4 * /ios3 * /ios2 * /ios1 * /ios0
io11 = ios4 * /ios3 * /ios2 *  ios1 * /ios0
io12 = ios4 * /ios3 *  ios2 *  ios1 * /ios0
io13 = ios4 * /ios3 *  ios2 *  ios1 *  ios0
io14 =/ios4 * /ios3 *  ios2 *  ios1 *  ios0
io15 =/ios4 *  ios3 *  ios2 *  ios1 *  ios0
io16 =/ios4 *  ios3 *  ios2 *  ios1 * /ios0
io17 =/ios4 *  ios3 *  ios2 * /ios1 * /ios0
io18 =/ios4 *  ios3 *  ios2 * /ios1 *  ios0
io19 =/ios4 *  ios3 * /ios2 * /ios1 *  ios0
io20 =/ios4 *  ios3 * /ios2 *  ios1 *  ios0
io21 =/ios4 * /ios3 * /ios2 *  ios1 *  ios0
io22 =/ios4 *  ios3 * /ios2 *  ios1 * /ios0
io23 =/ios4 * /ios3 *  ios2 *  ios1 * /ios0
io24 =/ios4 * /ios3 *  ios2 * /ios1 *  ios0
io32 =/ios4 * /ios3 * /ios2 *  ios1 * /ios0 ; used only on 1st Cx/Hx read.

io0  := ( /selio * /selcio * ciotrc) -> io1
      + ( /selio  * selcio ) -> io15
      + ( /selpmrom ) ->io15
      + ( time3 ) ->io15
```

Table A-1. DRAM Controller EPX780 PLDshell File (Sheet 5 of 12)

```
io1  := vcc -> io2
io2  := vcc -> io3
io3  := vcc -> io4
io4  := vcc -> io5
io5  := vcc -> io6
io6  := vcc -> io7
io7  := vcc -> io8
io8  := vcc -> io9
io9  := vcc -> io10
io10 := vcc -> io11
io11 := vcc -> io12
io12 := vcc -> io13
io13 := vcc -> io14
io14 := vcc -> io15
io15 := vcc -> io16
io16 := vcc -> io17
io17 := vcc -> io18
io18 := vcc -> io19
io19 := vcc -> io20
io20 := vcc -> io21
io21 := ( BOOTRD1 ) -> io32    ; Cx first read, loop & hold ready
      + (/BOOTRD1 ) -> io22    ; low until Cx asserts blast.
io22 := ( blast ) -> io15      ; Sx/Kx burst read
      + (/blast ) -> io23
io23 := vcc -> io24
io24 := vcc -> io0
io32 := ( blast ) -> io32
      + (/blast ) -> io23

        ;*********** Local bus Timeout *************
        ;8us minimum timeout on squall cycles.

STATE MEALY_MACHINE
DEFAULT_BRANCH HOLD_STATE
time0     =  t1 *  t0
time1     =  t1 * /t0
time2     = /t1 * /t0
time3     = /t1 *  t0


time0     := (/selsq0)              -> time1
time1     := (/readyin)             -> time0
      + ( refclk * /refclkd) -> time2
time2     := (/readyin)             -> time0
      + ( refclk * /refclkd) -> time3
time3     := (/readyin)             -> time0
```

Table A-1. DRAM Controller EPX780 PLDshell File (Sheet 6 of 12)

```
        ;*********** DRAM Control *****************
STATE MEALY_MACHINE
DEFAULT_BRANCH s0
s0    =  ref *  ras0 *  ras2 * /st2 * /st1  *  st0
s1    =  ref * /ras0 * /ras2 * /st2 * /st1  *  st0
s2    =  ref * /ras0 * /ras2 * /st2 * /st1  * /st0
s3    =  ref * /ras0 * /ras2 * /st2 *  st1  * /st0
s4    =  ref * /ras0 * /ras2 * /st2 *  st1  *  st0
s5    =  ref *  ras0 *  ras2 * /st2 * /st1  * /st0
s5a   =  ref *  ras0 *  ras2 *  st2 * /st1  * /st0
s6    = /ref *  ras0 *  ras2 * /st2 * /st1  *  st0
s7    = /ref *  ras0 *  ras2 * /st2 * /st1  * /st0
s8    = /ref * /ras0 * /ras2 * /st2 * /st1  * /st0
s9    = /ref * /ras0 * /ras2 *  st2 * /st1  * /st0
s10   = /ref * /ras0 * /ras2 *  st2 * /st1  *  st0

s0   := (/refpend)                                  -> s6
      + ( refpend * /ads * seldram */(pf0))         -> s1
      + ( refpend * /mempend       */(pf0))         -> s1
      + ( refpend * /ads * seldram * (pf0))         -> s2
      + ( refpend * /mempend       * (pf0))         -> s2
s1   := vcc      -> s2
s2   := (pf2 * /w_r)      -> s3
      + (pf2 *  w_r)      -> s4
      + (pf3 * /w_r)      -> s3
      + (pf3 *  w_r)      -> s4
      + (pf1)             -> s4
      + (pf0)             -> s4
s3   := vcc      -> s4
s4   := ( blast * /w_r * extend * (pf0))          -> s4   ;AP ok
      + ( blast * /w_r * extend * pf1)            -> s4
      + ( blast * /w_r * extend * pf2)            -> s3
      + ( blast * /w_r * extend * pf3)            -> s3
      + ( blast *  w_r )                          -> s3   ;AP ok
      + (/blast            * extend * (pf0))      -> s0   ;AP ok
      + (/blast            * extend * pf1)        -> s5
      + (/blast            * extend * pf2)        -> s5
      + (/blast *  w_r      * pf3)                -> s5a
      + (/blast * /w_r * extend * pf3)            -> s5
      + (/extend * /w_r )                         -> s4
s5a  := vcc      -> s5
s5   := vcc      -> s0
s6   := vcc      -> s7       ;refresh
s7   := vcc      -> s8       ;refresh
s8   := vcc      -> s9       ;refresh
s9   := vcc      -> s10      ;refresh
s10  := vcc      -> s5       ;refresh
```

```
st0.setf   = /rst  st1.rstf  = /rst   st2.rstf  = /rst
ras0.setf  = /rst  ras2.setf = /rst   ref.setf  = /rst

ios4.setf  = /rst  ios3.setf = /rst   ios2.setf = /rst
ios1.setf  = /rst  ios0.setf = /rst
t1.setf    = /rst  t0.setf   = /rst

irqddlk.clkf = pclk          boff.clkf = pclk
irqddlk.setf = /rst          boff.setf = /rst
```

Table A-1. DRAM Controller EPX780 PLDshell File (Sheet 8 of 12)

```
d0.clkf = pclk                  d1.clkf = pclk
d0.rstf = /rst                  d1.rstf = /rst

altREADY.clkf = pclk            altRDYEN.clkf = pclk
altREADY.setf = /rst            altRDYEN.rstf = /rst

 BOOTRD0.clkf = pclk
 BOOTRD1.clkf = pclk

        ;************ Deadlock Equations ***********
/altREADY := sD
           + sE * blast
 altREADY.trst = altRDYEN
 altRDYEN := sC + sD + sE
    /boff := Cx *  breqo
           + Cx * /breqo * /boff * pcihold

;************ Select Signals ***************
 /selio    := /ads * selb                              ;Bxxxxxxx
            + /ads * self * /swaprom                   ;Fxxxxxxx
            + /ads * sele *  swaprom                   ;Exxxxxxx
            + /ads * seld                              ;Dxxxxxxx
            + /ads * sel4_7 * /pciinstall
            + /ads * sel8_9 * /pciinstall
            + /selio * /(/blast * /ready) * rst

 /selcio := /ads   * selb */a19 *  a18                 ;b4000000
           + /selio * selb */a19 *  a18 * ready
 /seluart:= /ads   * selb */a19 * /a18                 ;b0000000
           + /selio * selb */a19 * /a18 * ready
 /selpp  := /ads   * selb * a19 * /a18                 ;b8000000
           + /selio * selb * a19 * /a18 * ready

 /selpmrom:= swaprom * /ads      * self      * rst ;f0000000
           + swaprom * /selpmrom * self   * ready * rst
           + swaprom * /ads      * sel0_3      * rst ;00000000
           + swaprom * /selpmrom * sel0_3 * ready * rst
           +/swaprom * /ads      * sele      * rst ;e0000000
           +/selpmrom */(/blast * /ready)    * rst

 /selfrom0:= swaprom * /ads   * sele   */a18          ;e0000000
           + swaprom * /selio * sele   */a18 * ready
           +/swaprom * /ads   * self   */a18          ;f0000000
           +/swaprom * /selio * self   */a18 * ready
           +/swaprom * /ads   * sel0_3 */a18          ;00000000
           +/swaprom * /selio * sel0_3 */a18 * ready
```

Table A-1. DRAM Controller EPX780 PLDshell File (Sheet 9 of 12)

```
/selfrom1:= swaprom * /ads    * sele    * a18          ;e0040000
          + swaprom * /selio * sele    * a18 * ready
          +/swaprom * /ads    * self    * a18          ;f0040000
          +/swaprom * /selio * self    * a18 * ready
          +/swaprom * /ads    * sel0_3 * a18          ;00040000
          +/swaprom * /selio * sel0_3 * a18 * ready
          + /selfrom1 * BOOTRD1 * blast * rst

/selsq0   := /ads * selc * rst                        ;c0000000
          + /selsq0 * /(/blast * /readyin) * rst

BOOTRD0 := /rst                    * /pt2 * pt1       ; set if Cx or Hx
         +  rst * BOOTRD0 *  ads   * /pt2 * pt1       ; hold til 1st ads

BOOTRD1 :=  rst * BOOTRD0 * /ads   * /pt2 * pt1       ; set at ads
         +  rst * BOOTRD1 *  blast * /pt2 * pt1       ; hold til 1st blast

          ;************I/O control equatons *********
   /zrd := /rst
         + /selcio * /w_r * io2 * rst
         + /selcio * /w_r * io3 * rst
         + /selcio * /w_r * /zrd * ready * rst

   /zwr := /rst
         + /selcio *  w_r * io2 * rst
         + /selcio *  w_r * io3 * rst
         + /selcio *  w_r * /zwr * ready * rdyen * rst

   /ior := /selio * selcio * /w_r * ready * rst
         + /selpmrom       * /w_r * ready * rst
         + /selio * selcio * /w_r * /ready * BOOTRD1 * blast * rst
         + /selpmrom       * /w_r * /ready * BOOTRD1 * blast * rst

   /iow := /selio * selcio *  w_r * ready * rdyen * rst
         + /selpmrom       *  w_r * ready * rdyen * rst

/irquartn= irquart
          ;************ Configuration Terms **********
 sx      = /pt2 * /pt1 * /pt0 * /holda                ;i960Sx
 selmux  =  freq2 */freq1 *  freq0                    ; 33MHz
         +  freq2 * freq1 * /freq0                    ; 40MHz
         +  freq2 * freq1 *  freq0                    ; 50MHz
          ;************DRAM Control ****************
   /mempend  :=  mempend * /ads * seldram * rst
              + /mempend * /(/blast * /ready) * rst ;hold till rdy

 mempend.setf = /rst
/mux         := /ras0 * rst
/muxalt      := /ras0 * rst
 mux.clkf     = /pclk
```

Table A-1. DRAM Controller EPX780 PLDshell File (Sheet 10 of 12)

```
mux.trst      = /selmux      ;MUX tri-stated when operating @ >=33MHz
muxalt.clkf   =  pclk
muxalt.trst   =  selmux

/casen        := s0 * /ads * seldram * (pf0) * refpend
                 + s0 * /mempend * (pf0)  * refpend
                 + s1
                 + s2 * /w_r * pf2
                 + s2 * /w_r * pf3
                 + s2 * ((pf0) + (pf1)) * /w_r
                 + s3 * /w_r
                 + s4 * blast
                 + s4 * /extend * /w_r * /blast


/incbank      := s0 * /ads * seldram * (pf0) * refpend * extend ; JX ok
                 + s0 * /mempend * (pf0) * refpend * extend       ; JX ok
                 + s1 * /((pf2) + (pf3)) * /w_r * blast * extend ; JX ok
                 + s1 *  w_r * blast * extend                    ;   "
                 + s2 * /w_r * blast * extend                    ;   "
                 + s3 * /((pf2) + (pf3)) * /w_r * blast * extend ; 1st term
                                                                 ; =1 w/
                                                                 ; pf0=1
                 + s4 * blast       * extend                     ;   "

/ready        := s2 * (pf0)
                 + s2 * pf1
                 + s2 * pf2 * w_r
                 + s2 * pf3 * w_r
                 + s3
                 + s4 * (pf0) * /w_r * blast
                 + s4 * pf1 * /w_r * blast
                 + s4 * /w_r * /blast * /extend
                 + io21
                 + io32

ready.trst    = /rdyen

/rdyen        := s1 + s2 + s3 + s4
                 + io20 + io21 + io22 + io32

/dwe          :=  w_r * ref

ma0a.d :=  ras0 * /a3 *  a2    ; a[3:2] = 01
         +  ras0 *  a3 * /a2    ;   "    = 10
         + /ras0 * /ma0b *  banksel * bss0 * /ready
         + /ras0 *  ma0a * /banksel
         + /ras0 *  ma0a * /bss0
         + /ras0 *  ma0a *  ready
```

```
    ma0a.trst  = /muxin    ;output enable active with MUX or MUXALT low
    ma0b.trst  = /muxin    ;muxin low when MUX or MUXALT is low

            ;************ DRAM Bank Select *************

  /banksel.d:= /ads *  mempend * /sx * /a2 * rst                    ; JX ok
            + /ads *  mempend *  sx * /a2 * /a1 * rst
            + /ads *  mempend *  sx * /a2 *  a1 * rst
            + /mempend *  incbank * /banksel * rst                  ;hold JX ok
            + /mempend * /incbank * /banksel * rst * /extend        ;hold
            + /mempend * /incbank * /banksel * /bss0 * rst * extend
                                                                    ;bs0->bs1
            + /mempend * /incbank *  banksel *  bss0 * rst * extend ; JX ok
                                   ; /\                             ;bs3->bs0
                                     ||
                                  ; oscillates with incbank low
;bs3->bs1 /sx

;bs3->bs1 /sx
  /bss0.d   := /ads * sx * /a2 * /a1 * rst
            + /ads * sx *  a2 * /a1 * rst
            +  ads * sx *  incbank * /bss0 * rst                    ;hold
            +  ads * sx * /incbank * /bss0 * rst * /extend          ;hold
            +  ads * sx * /incbank * /banksel * bss0 * rst * extend
                                                                    ;bs1 ->bs2
            +  ads * sx * /incbank *  banksel * bss0 * rst * extend
                                                                    ;bs3 ->bs0
    banksel.setf = /rst   bss0.setf = /rst

            ;************ DRAM Refresh *****************
    /refsync  := refc5                  ;sync 4MHz count w/ pclock
    /refclk   := refsync
    /refclkd  := /refclk
    /refpend  :=  refclk * /refclkd * rst   ;set
               + /refpend *  ref * rst      ;reset with ref

    refc0.t   := vcc                                    ;refresh counter 16us
    refc1.t   := refc0
    refc2.t   := refc0 * refc1
    refc3.t   := refc0 * refc1 * refc2
    refc4.t   := refc0 * refc1 * refc2 * refc3
```

Table A-1. DRAM Controller EPX780 PLDshell File (Sheet 12 of 12)

```
refc5.t   := refc0 * refc1 * refc2 * refc3 * refc4
refc0.aclk = clk4  refc1.aclk = clk4  refc2.aclk = clk4
refc3.aclk = clk4  refc4.aclk = clk4  refc5.aclk = clk4
refsync.clkf=pclk  refclk.clkf= pclk  refclkd.clkf=pclk
refpend.clkf=pclk  ref.clkf   = pclk

       ;************ Arbitration *****************
 hold            := pcihold * /holda * lock  ;hold may not be asserted
          + pcihold *  holda                 ;if lock is active, but
          + /sqbr0  * /holda * lock          ;lock maybe asserted by
          + /sqbr0  *  holda                 ;SQUALL masters once they
                                             ;are granted the bus.
/sqbg0           := holda * hold * /sqbr0 * /pcihlda
          + /sqbg0 * /sqbr0 * holda

 pcihlda := holda * hold * pcihold * sqbr0 * sqbg0
          + pcihlda * pcihold * holda
          + pcihold * /boff

       ;************ 4 MHz clock *****************
 clk8.t       := vcc
 clk8.clkf    = clk16
 clk4.t       := vcc
 clk4.aclk    = clk8
       ;************ CIO T recovery **************
       ;Delay 1000ns for trc on the 8536
/ciotrc       := /selcio * /ready    * rst       ;set
          + /ciotrc *   synctrc * rst            ;hold/reset

/synctrc      := /ciotrc *  /count1 *      count0 * rst ;assert

 ciotrc.clkf   = pclk
 synctrc.clkf  = pclk

/count1       := /ciotrc *  count1 * /count0 * rst ;10 -> 00
                 + /ciotrc * /count1 * /count0 * rst ;00 -> 01

/count0       := /ciotrc *  count1 *  count0 * rst ;11 -> 10
                 + /ciotrc *  count1 * /count0 * rst ;10 -> 00
 count1.aclk  = clk4
 count0.aclk  = clk4
 count1.setf  = /synctrc *  ciotrc
 count0.setf  = /synctrc *  ciotrc
```

Table A-2. DRAM Controller PAL20V8 PLDshell File (Sheet 1 of 2)

```
TITLE          Private DRAM Control
PATTERN
REVISION       D600A
AUTHOR         J. Smith
COMPANY        Cyclone
DATE           06-14-94
CHIP           CASEN   PAL20V8

PIN     1      pclk              ;INP:      system clock
PIN     2      ready             ;INP:      ;plcc pin 3
PIN     3      casen             ;INP:      ;plcc pin 4
PIN     4      banksel           ;INP:      ;plcc pin 5
PIN     5      blast             ;INP:      ;plcc pin 6
PIN     6      be3               ;INP:      ;plcc pin 7
PIN     7      be2               ;INP:      ;plcc pin 9
PIN     8      be1               ;INP:      ;plcc pin 10
PIN     9      be0               ;INP:      ;plcc pin 11
PIN     10     ref               ;INP:      ;plcc pin 12
PIN     11     nc                ;INP:      ;plcc pin 13
PIN     12     gnd                          ;plcc pin 14
PIN     13     oe                           ;plcc pin 16
PIN     14     nc                ;OUT:      ;plcc pin 17
PIN     15     casb0     REG     ;OUT:      ;plcc pin 18
PIN     16     casb1     REG     ;OUT:      ;plcc pin 19
PIN     17     casb2     REG     ;OUT:      ;plcc pin 20
PIN     18     casb3     REG     ;OUT:      ;plcc pin 21
PIN     19     casa0     REG     ;OUT:      ;plcc pin 23
PIN     20     casa1     REG     ;OUT:      ;plcc pin 24
PIN     21     casa2     REG     ;OUT:      ;plcc pin 25
PIN     22     casa3     REG     ;OUT:      ;plcc pin 26
PIN     23     nc                           ;plcc pin 27
PIN     24     vcc
```

```
        + /ref
/casb1  := /casen  * /be1 *   banksel * /(/blast * /ready)
        + /ref
/casb2  := /casen  * /be2 *   banksel * /(/blast * /ready)
        + /ref
/casb3  := /casen  * /be3 *   banksel * /(/blast * /ready)
        + /ref
```

APPENDIX B
PLD EQUATIONS

Table B-1 contains the PLD equations used to build the SAR controller, implemented in an AMD* MACH210* PLD. The PLD equations were created in MACHXL.

Table B-1. SAR Controller MACH210, MACHXL File (Sheet 1 of 6)

```
     TITLE  NEC-SAR (uPD98401) Control
   PATTERN  H601
  REVISION  A
    AUTHOR  Joe Niedermeyer
   COMPANY  Cyclone
      DATE  6-06-95

      CHIP  H601  MACH210  ;(-12nS tPD req'd for SARRDYn)


PIN   35  PCLK             ;INP: system clock.
PIN    7  RESETn           ;INP: system reset.
PIN   19  SELSQn           ;INP: squall region decode.
PIN   14  SARATTNn         ;INP: SAR bus request.
PIN    3  SQBGNTn          ;INP: squall bus grant.
PIN   13  SIZE2            ;INP: SAR burst length MSB.
PIN   11  SIZE1            ;INP: SAR burst length bit.
PIN   10  SIZE0            ;INP: SAR burst length LSB.
PIN    5  SARDWRn          ;INP: SAR dr/w* (master rd/wr*).
PIN   33  PHYCSn           ;INP: FRAMER chip select.
PIN   32  PHYWRn           ;INP: FRAMER read/write*.

PIN    2  READYpinn  REG   ;OUT: i960 ready*.
PIN    6  SARASELn   REG   ;OUT: SAR asel*.
PIN   41  SARSELn    REG   ;OUT: SAR sel*.
PIN   36  SARSWRn    REG   ;OUT: SAR sr/w* (slave rd/wr*).
PIN   25  SQBREQn    REG   ;OUT: squall bus request.
PIN   30  SARGNTn    REG   ;OUT: SAR bus grant.
PIN    4  BLASTpinn  REG   ;OUT: blast* when SAR is bus master.
PIN   16  ADSpinn    REG   ;OUT: ads* when SAR is bus master.
PIN   21  CPUWRpin   REG   ;OUT: w/r* when SAR is bus master.
PIN   24  SARRDYn    COMB  ;OUT: SAR rdy*.
PIN   18  BE0n       COMB  ;OUT: be0* when SAR is bus master.
PIN   20  BE1n       COMB  ;OUT: be1* when SAR is bus master.
PIN   28  BE2n       COMB  ;OUT: be2* when SAR is bus master.
PIN   38  BE3n       COMB  ;OUT: be3* when SAR is bus master.
PIN   26  AOEBAn     REG   ;OUT: ADDR XCVR control, master direction.
PIN   40  ACLKENn    REG   ;OUT: ADDR XCVR clock enable, master direction.
PIN   42  AOEABn     REG   ;OUT: ADDR XCVR control, slave direction.
PIN   43  DOEABn     REG   ;OUT: DATA XCVR control, slave write direction.
PIN   17  DOEBAn     REG   ;OUT: DATA XCVR control, slave read direction.
PIN   15  DCLKENn    COMB  ;OUT: DATA XCVR clock enable, master read dir.
PIN   37  DLEABn     REG   ;OUT: DATA XCVR latch enable, master write dir.
PIN   39  PHYALE     REG   ;OUT: FRAMER address latch enable.
```

Table B-1. SAR Controller MACH210, MACHXL File (Sheet 2 of 6)

```
NODE  2  READYn      REG  PAIR  READYpinn
NODE 28  ADSn        REG  PAIR  ADSpinn
NODE  6  BLASTn      REG  PAIR  BLASTpinn
NODE 18  CPUWR       REG  PAIR  CPUWRpin

NODE 23  RDYENn      REG  ;BRY: ready pin enable.
NODE 11  CNT3        REG  ;BRY: burst count MSB.
NODE  7  CNT2        REG  ;BRY: burst count bit.
NODE  4  CNT1        REG  ;BRY: burst count bit.
NODE 15  CNT0        REG  ;BRY: burst count LSB.
NODE  3  SARRDY0     REG  ;BRY: ready*, one clock delayed.

NODE  14,39  slv[1..0]  REG  ;BRY: slave states.
NODE  35,22  mst[1..0]  REG  ;BRY: master states.

;*********************************;
;***  Slave State Assignments  ***;
;*********************************;

STRING   WAIT  ' #b11 '
STRING      A  ' #b01 '
STRING      B  ' #b00 '
STRING      C  ' #b10 '

STRING  sWAIT  '  slv[1] *  slv[0] '
STRING     sA  ' /slv[1] *  slv[0] '
STRING     sB  ' /slv[1] * /slv[0] '
STRING     sC  '  slv[1] * /slv[0] '

;*********************************;
;***  Master State Assignments  ***;
;*********************************;

STRING     M0  ' #b11 '
STRING     M1  ' #b01 '
STRING     M2  ' #b00 '
STRING     M3  ' #b10 '

STRING    sM0  '  mst[1] *  mst[0] '
STRING    sM1  ' /mst[1] *  mst[0] '
STRING    sM2  ' /mst[1] * /mst[0] '
STRING    sM3  '  mst[1] * /mst[0] '

GROUP  ff  slv[1..0]  mst[1..0]  READYn  RDYENn
           SARASELn  SARSELn  SARSWRn  SQBREQn SARGNTn
           AOEABn AOEBAn ACLKENn DOEABn DOEBAn DLEABn
           ADSn CPUWR SARRDY0 BLASTn PHYALE
           CNT3 CNT2 CNT1 CNT0

EQUATIONS

   ff.CLKF  = PCLK
   ff.RSTF  = GND
   ff.SETF  = GND
```

Table B-1. SAR Controller MACH210, MACHXL File (Sheet 3 of 6)

```
;*********************************;
;***  Slave State Transitions  ***;
;*********************************;

IF (/RESETn)
THEN  BEGIN  slv[1..0] = WAIT  END
ELSE  BEGIN  CASE  (slv[1..0])
      BEGIN
      WAIT:  BEGIN IF  (SELSQn)
                   THEN BEGIN  slv[1..0] = WAIT  END
                   ELSE BEGIN  slv[1..0] = A     END
             END
         A:  BEGIN  slv[1..0] = B     END
         B:  BEGIN  slv[1..0] = C     END
         C:  BEGIN  slv[1..0] = WAIT  END
      END
      END

;*************************;
;***  Ready Generation  ***;
;*************************;

    READYn := /( sB )

 READYpinn := { READYn }

 READYpinn.TRST = /RDYENn

   /RDYENn := /SELSQn

;*****************************;
;***  SAR Slave Equations  ***;
;*****************************;

 /SARASELn := sWAIT * /SELSQn

  /SARSELn := sWAIT * /SELSQn
            + sA

  /SARSWRn := sWAIT * /SELSQn * CPUWRpin
```

```
    /DOEABn :=  sA *  CPUWRpin                          ; SAR Slave Write
             + sM3 *  SARDWRn                           ; SAR Master Read

    /DOEBAn :=  sB * /CPUWRpin                          ; SAR Slave Read
             + sM3 * /SARDWRn                           ; SAR Master Write

   /DCLKENn  = /READYpinn * /SQBGNTn * SARDWRn          ; SAR Master Read Only

    /DLEABn := sM3 *  SARDWRn                           ; SAR Master Read Only

  ;*********************************;
  ;***  Master State Transitions  ***;
  ;*********************************;

  IF (/RESETn)
  THEN  BEGIN  mst[1..0] = M0  END
  ELSE  BEGIN  CASE  (mst[1..0])
        BEGIN
          M0:  BEGIN IF  (/SQBGNTn * /SARATTNn)
                       THEN BEGIN  mst[1..0] = M1  END
                       ELSE BEGIN  mst[1..0] = M0  END
               END
          M1:  BEGIN  mst[1..0] = M2    END
          M2:  BEGIN  mst[1..0] = M3    END
          M3:  BEGIN IF  (/BLASTn * /READYpinn)
                       THEN BEGIN  mst[1..0] = M0  END
                       ELSE BEGIN  mst[1..0] = M3  END
               END
        END
        END

  ;*******************************;
  ;***  Arbitration Equations  ***;
  ;*******************************;

    /SQBREQn := /SARATTNn * RESETn

    /SARGNTn := /SQBGNTn * /SQBREQn * /SARATTNn * RESETn
```

Table B-1. SAR Controller MACH210, MACHXL File (Sheet 5 of 6)

```
;*************************;
;***  Blast Generation  ***;
;*************************;

; Burst counter needed to generate BLASTn.
; SAR indicates burst length (encoded) on SIZE<2,1,0> pins.
; Counter loaded with (SIZE - 1) at ADS*.
; Counter decrements with each READY*.
; BLASTn asserted when (CNT = 1) and (READY* = 0).

CNT3 := /ADSn *  SIZE2 * /SIZE1 * /SIZE0                          ; LOAD
      +  ADSn *  READYpinn *  CNT3                                ; HOLD
      +  ADSn * /READYpinn *  CNT3 *  CNT2 *  CNT1 *  CNT0        ; DOWN
      +  ADSn * /READYpinn *  CNT3 *  CNT2 *  CNT1 * /CNT0        ; DOWN
      +  ADSn * /READYpinn *  CNT3 *  CNT2 * /CNT1 *  CNT0        ; DOWN
      +  ADSn * /READYpinn *  CNT3 *  CNT2 * /CNT1 * /CNT0        ; DOWN
      +  ADSn * /READYpinn *  CNT3 * /CNT2 *  CNT1 *  CNT0        ; DOWN
      +  ADSn * /READYpinn *  CNT3 * /CNT2 *  CNT1 * /CNT0        ; DOWN
      +  ADSn * /READYpinn *  CNT3 * /CNT2 * /CNT1 *  CNT0        ; DOWN
      +  ADSn * /READYpinn * /CNT3 * /CNT2 * /CNT1 * /CNT0        ; DOWN

CNT2 := /ADSn *  SIZE2 * /SIZE1 * /SIZE0                          ; LOAD
      + /ADSn * /SIZE2 *  SIZE1 *  SIZE0                          ; LOAD
      +  ADSn *  READYpinn *  CNT2                                ; HOLD
      +  ADSn * /READYpinn *  CNT2 *  CNT1 *  CNT0                ; DOWN
      +  ADSn * /READYpinn *  CNT2 *  CNT1 * /CNT0                ; DOWN
      +  ADSn * /READYpinn *  CNT2 * /CNT1 *  CNT0                ; DOWN
      +  ADSn * /READYpinn * /CNT2 * /CNT1 * /CNT0                ; DOWN

CNT1 := /ADSn *  SIZE2 * /SIZE1 * /SIZE0                          ; LOAD
      + /ADSn * /SIZE2 *  SIZE1 *  SIZE0                          ; LOAD
      + /ADSn * /SIZE2 *  SIZE1 * /SIZE0                          ; LOAD
      +  ADSn *  READYpinn *  CNT1                                ; HOLD
      +  ADSn * /READYpinn *  CNT1 *  CNT0                        ; DOWN
      +  ADSn * /READYpinn * /CNT1 * /CNT0                        ; DOWN

CNT0 := /ADSn *  SIZE2 * /SIZE1 * /SIZE0                          ; LOAD
      + /ADSn * /SIZE2 *  SIZE1 *  SIZE0                          ; LOAD
      + /ADSn * /SIZE2 *  SIZE1 * /SIZE0                          ; LOAD
      + /ADSn * /SIZE2 * /SIZE1 *  SIZE0                          ; LOAD
      +  ADSn *  READYpinn *  CNT0                                ; HOLD
      +  ADSn * /READYpinn * /CNT0                                ; DOWN

   BLASTn := /( /ADSn * /SIZE2 * /SIZE1 * /SIZE0 * RESETn
              + /READYpinn * /CNT3 * /CNT2 * /CNT1 * CNT0 * RESETn
              + /BLASTn    * READYpinn             * RESETn )

 BLASTpinn := { BLASTn }
BLASTpinn.TRST = /SQBGNTn
```

Table B-1. SAR Controller MACH210, MACHXL File (Sheet 6 of 6)

```
;*****************************;
;***  SAR Master Equations  ***;
;*****************************;

      ADSn := /( sM2 )
   ADSpinn := { ADSn }
ADSpinn.TRST = /SQBGNTn

    CPUWR  := /SARDWRn
 CPUWRpin  := { CPUWR }
CPUWRpin.TRST = /SQBGNTn

  /SARRDY0 := /READYpinn

  /SARRDYn  =   /SARRDY0 * /SQBGNTn *  SARDWRn        ; READs pipelined
            + /READYpinn * /SQBGNTn * /SARDWRn        ; WRITEs normal

    /BE0n  = VCC
    /BE1n  = VCC
    /BE2n  = VCC
    /BE3n  = VCC

 BE0n.TRST  = /SQBGNTn
 BE1n.TRST  = /SQBGNTn
 BE2n.TRST  = /SQBGNTn
 BE3n.TRST  = /SQBGNTn

;*******************************;
;***  Physical I/F Equations ***;
;*******************************;

; SAR only holds a valid address for the 1st 6 clocks of a PHY read cycle.
; PHY write cycle works OK, as-is.
; PHY has an active HI ALE input available.
; ALE is held HI for write cycles and pulled LO during reads to latch addr.

   PHYALE := /( /PHYCSn *  PHYWRn )

;*********************;
;***  End Of H601  ***;
;*********************;
```

intel®

2

Memories and Peripherals

Peripherals

28F016XS
16-MBIT (1 MBIT X 16, 2 MBIT X 8) SYNCHRONOUS FLASH MEMORY

- **Effective Zero Wait-State Performance up to 33 MHz**
  - **Synchronous Pipelined Reads**
- **SmartVoltage Technology**
  - **User-Selectable 3.3V or 5V V_{CC}**
  - **User-Selectable 5V or 12V V_{PP}**
- **0.33 MB/sec Write Transfer Rate**
- **Configurable x8 or x16 Operation**
- **56-Lead TSOP and SSOP Type I Package**
- **Backwards-Compatible with 28F008SA Command-Set**
- **2 µA Typical Deep Power-Down**
- **1 mA Typical Active I_{CC} Current in Static Mode**
- **16 Separately-Erasable/Lockable 128-Kbyte Blocks**
- **1 Million Erase Cycles per Block**
- **State-of-the-Art 0.6 µm ETOX™ IV Flash Technology**

Intel's 28F016XS 16-Mbit flash memory is a revolutionary architecture which is the ideal choice for designing truly revolutionary high-performance products. Combining very high read performance with the intrinsic nonvolatility of flash memory, the 28F016XS eliminates the traditional redundant memory paradigm of shadowing code from a slow nonvolatile storage source to a faster execution memory, such as DRAM, for improved system performance. The innovative capabilities of the 28F016XS enable the design of direct-execute code and mass storage data/file flash memory systems.

The 28F016XS is the highest performance high-density nonvolatile read/program flash memory solution available today. Its synchronous pipelined read interface, flexible V_{CC} and V_{PP} voltages, extended cycling, fast program and read performance, symmetrically-blocked architecture, and selective block locking provide a highly flexible memory component suitable for resident flash component arrays on the system board or SIMMs. The synchronous pipelined interface and x8/x16 architecture of the 28F016XS allow easy interface with minimal glue logic to a wide range of processors/buses, providing effective zero wait-state read performance up to 33 MHz. The 28F016XS's dual read voltage allows the same component to operate at either 3.3V or 5.0V V_{CC}. Programming voltage at 5V V_{PP} minimizes external circuitry in minimal-chip, space critical designs, while the 12.0V V_{PP} option maximizes program/erase performance. Its high read performance combined with flexible block locking enable both storage and execution of operating systems/application software and fast access to large data tables. The 28F016XS is manufactured on Intel's 0.6 µm ETOX IV process technology.

1.0 INTRODUCTION

The documentation of the Intel 28F016XS Flash memory device includes this datasheet, a detailed user's manual, a number of application notes and design tools, all of which are referenced in Appendix B.

The datasheet is intended to give an overview of the chip feature-set and of the operating AC/DC specifications. The *16-Mbit Flash Product Family User's Manual* provides complete descriptions of the user modes, system interface examples and detailed descriptions of all principles of operation. It also contains the full list of software algorithm flowcharts, and a brief section on compatibility with the Intel 28F008SA.

Significant 28F016XS feature revisions occurred between datasheet revisions 290532-001 and 290532-002. These revisions center around removal of the following features:

- All page buffer operations (read, write, programming, Upload Device Information)
- Command queuing
- Software Sleep and Abort
- Erase all Unlocked Blocks and Two-Byte Write
- RY/BY# Configuration options

In addition, a significant 28F016XS change occurred between datasheet revisions 290532-002 and 290532-003. This change centers around the addition of a 3/5# pin to the device's pinout configuration. Figures 2 and 3 show the 3/5# pin assignment for the TSOP and SSOP Type I packages.

Intel recommends that all customers obtain the latest revisions of 28F016XS documentation.

1.1 Product Preview

The 28F016XS is a high-performance, 16-Mbit (16,777,216-bit) block erasable nonvolatile random access memory organized as either 1 Mword x 16 or 2 Mbyte x 8, subdivided into even and odd banks. Address A_1 makes the bank selection. The 28F016XS includes sixteen 128-Kbyte (131,072 byte) blocks or sixteen 64-Kword (65,536 word) blocks. Chip memory maps for x8 and x16 modes are shown in Figures 4 and 5.

The implementation of a new architecture, with many enhanced features, will improve the device operating characteristics and result in greater product reliability and ease-of-use as compared to other flash memories. Significant features of the 28F016XS as compared to previous asynchronous flash memories include:

- Synchronous Pipelined Read Interface
- Significantly Improved Read and Program Performance
- SmartVoltage Technology
  - — Selectable 3.3V or 5.0 V_{CC}
  - — Selectable 5.0V or 12.0 V_{PP}
- Block Program/Erase Protection

The 28F016XS's synchronous pipelined interface dramatically raises read performance far beyond previously attainable levels. Addresses are synchronously latched and data is read from a 28F016XS bank every 30 ns (5V V_{CC}, SFI Configuration = 2). This capability translates to zero wait-state reads at clock rates up to 33 MHz at 5V V_{CC}, after an initial address pipeline fill delay and assuming even and odd banks within the flash memory are alternately accessed. Data is latched and driven valid 20 ns (t_{CHQV}) after a rising CLK edge. The 28F016XS is capable of operating up to 50 MHz (5V V_{CC}); its programmable SFI Configuration enables system design flexibility, optimizing the 28F016XS to a specific system clock frequency. See Section 4.9, SFI Configuration Table, for specific SFI Configurations for given operating frequencies.

The SFI Configuration optimizes the 28F016XS for a wide range of system operating frequencies. The default SFI Configuration is 4, which allows system boot from the 28F016XS at any frequency up to 50 MHz at 5V V_{CC}. After initiating an access, data is latched and begins driving on the data outputs after a CLK count corresponding to the SFI Configuration has elapsed. The 28F016XS will hold data valid until CE# or OE# is deactivated or a CLK count corresponding to the SFI Configuration for a subsequent access has elapsed.

The CLK and ADV# inputs, new to the 28F016XS in comparison to previous flash memories, control address latching and device synchronization during read operations. The CLK input controls the device latencies, times out the SFI Configuration counter and synchronizes data outputs. ADV# indicates the

presence of a valid address on the 28F016XS address inputs. During read operations, addresses are latched and accesses are initiated on a rising CLK edge in conjunction with ADV# low. Both CLK and ADV# are ignored by the 28F016XS during command/data write sequences.

The 28F016XS incorporates SmartVoltage technology, providing V_{CC} operation at both 3.3V and 5.0V and program and erase capability at V_{PP} = 12.0V or 5.0V. Operating at V_{CC} = 3.3V, the 28F016XS consumes less than one half the power consumption at 5.0V V_{CC}, while 5.0V V_{CC} provides highest read performance capability. V_{PP} operation at 5.0V eliminates the need for a separate 12.0V converter, while the V_{PP} = 12.0V option maximizes program/erase performance. In addition to the flexible program and erase voltages, the dedicated V_{PP} gives complete code protection with $V_{PP} \leq V_{PPLK}$.

A 3/5# input pin configures the device's internal circuitry for optimal 3.3V or 5.0V read/program operation.

A Command User Interface (CUI) serves as the system interface between the microprocessor or microcontroller and the internal memory operation.

Internal Algorithm Automation allows program and block erase operations to be executed using a Two-Write command sequence to the CUI in the same way as the 28F008SA 8-Mbit FlashFile™ memory.

Software locking of memory blocks is an added feature of the 28F016XS as compared to the 28F008SA. The 28F016XS provides selectable block locking to protect code or data such as direct-executable operating systems or application code. Each block has an associated nonvolatile lock-bit which determines the lock status of the block. In addition, the 28F016XS has a master Write Protect pin (WP#) which prevents any modifications to memory blocks whose lock-bits are set.

Writing of memory data is performed in either byte or word increments, typically within 6 µs at 12.0V V_{PP}, which is a 33% improvement over the 28F008SA. A block erase operation erases one of the 16 blocks in typically 1.2 sec, independent of the other blocks.

Each block can be written and erased a minimum of 100,000 cycles. Systems can achieve one million Block Erase Cycles by providing wear-leveling algorithms and graceful block retirement. These techniques have already been employed in many flash file systems and hard disk drive designs.

All operations are started by a sequence of Write commands to the device. Three Status Registers (described in detail later in this datasheet) and a RY/BY# output pin provide information on the progress of the requested operation.

The following Status Registers are used to provide device and WSM operation information to the user:

- A Compatible Status Register (CSR) which is 100% compatible with the 28F008SA FlashFile memory Status Register. The CSR, when used alone, provides a straightforward upgrade capability to the 28F016XS from a 28F008SA-based design.
- A Global Status Register (GSR) which also informs the system of overall Write State Machine (WSM) status.
- 16 Block Status Registers (BSRs) which provide block-specific status information such as the block lock-bit status.

The GSR and BSR memory maps for Byte-Wide and Word-Wide modes are shown in Figures 5 and 6.

The 28F016XS incorporates an open drain RY/BY# output pin. This feature allows the user to OR-tie many RY/BY# pins together in a multiple memory configuration such as a Resident Flash Array.

The 28F016XS also incorporates a dual chip-enable function with two input pins, CE_0# and CE_1#. These pins have exactly the same functionality as the regular chip-enable pin, CE#, on the 28F008SA. For minimum chip designs, CE_1# may be tied to ground and system logic may use CE_0# as the chip enable input. The 28F016XS uses the logical combination of these two signals to enable or disable the entire chip. Both CE_0# and CE_1# must be active low to enable the device. If either one becomes inactive, the chip will be disabled. This feature, along with the open drain RY/BY# pin, allows the system designer to reduce the number of control pins used in a large array of 16-Mbit devices.

the active current when the device is in static mode of operation (addresses not switching). In APS mode, the typical I_{CC} current is 1 mA at 5.0V (3 mA at 3.3V).

A deep power-down mode of operation is invoked when the RP# (called PWD# on the 28F008SA) pin transitions low. This mode brings the device power consumption to less than 2.0 µA, typically, and levels. In this mode, the device typically draws an I_{CC} standby current of 70 µA at 5V V_{CC}.

The 28F016XS is available in 56-Lead, 1.2 mm thick, 14 mm x 20 mm TSOP and 1.8 mm thick, 16 mm x 23.7 mm SSOP Type I packages. The form factor and pinout of these two packages allow for very high board layout densities.

2.0 ADDITIONAL INFORMATION

Order Number	Document/Tool
297372	*16-Mbit Flash Product Family User's Manual*
292147	*AP-398 Designing with the 28F016XS*
292146	*AP-600 Performance Benefits and Power/Energy Savings of 28F016XS-Based System Designs*
292163	*AP-610 Flash Memory In-System Code and Data Update Techniques*
292165	*AB-62 Compiled Code Optimizations for Flash Memories*
297500	*Interfacing the 28F016XS to the i960® Microprocessor Family*
297504	*Interfacing the 28F016XS to the Intel486™ Microprocessor Family*
294016	*ER-33 ETOX™ Flash Memory Technology—Insight to Intel's Fourth Generation Process Innovation*
297508	FLASHBuilder Utility
Contact Intel/Distribution Sales Office	28F016XS Benchmark Utility
Contact Intel/Distribution Sales Office	Flash Cycling Utility
Contact Intel/Distribution Sales Office	28F016XS iBIS Model
Contact Intel/Distribution Sales Office	28F016XS VHDL Model
Contact Intel/Distribution Sales Office	28F016XS TimingDesigner* Library Files
Contact Intel/Distribution Sales Office	28F016XS Orcad/Viewlogic Schematic Symbols

This complete document is available from Intel's World Wide Website and/or U.S. Literature Center:
World Wide Website: http://www.Intel.com
U. S. Literature Center: 800-548-4725
In other geographies, please contact your local sales office.

2.0 ADDITIONAL INFORMATION

Order Number	Document/Tool
297372	16-Mbit Flash Product Family User's Manual
292147	AP-398 Designing with the 28F016XS
292149	AP-600 Performance Benefits and PowerEnergy Savings of 28F016XS-Based System Designs
292163	AP-610 Flash Memory In-System Code and Data Update Techniques
292165	AB-62 Compiled Code Optimizations for Flash Memories
297500	Interfacing the 28F016XS to the i960® Microprocessor Family
297504	Interfacing the 28F016XS to the Intel486™ Microprocessor Family
296018	ER-33 ETOX™ Flash Memory Technology—Insight to Intel's Fourth Generation Process Innovation
297505	FLASHBuilder Utility
Contact Intel/Distribution Sales Office	28F016XS Benchmark Utility
Contact Intel/Distribution Sales Office	Flash Cycling Utility
Contact Intel/Distribution Sales Office	28F016XS IBIS Model
Contact Intel/Distribution Sales Office	28F016XS VHDL Model
Contact Intel/Distribution Sales Office	28F016XS ThinkingDesigner™ Library Files
Contact Intel/Distribution Sales Office	28F016XS OrcadViewlogic Schematic Symbols

This complete document is available from Intel's World Wide Website and/or U.S. Literature Center:
World Wide Website: http://www.intel.com
U.S. Literature Center: 800-548-4725
In other geographies, please contact your local sales office.

28F016XD
16-MBIT (1 MBIT X 16)
DRAM-INTERFACE FLASH MEMORY

- 85 ns Access Time (t_{RAC})
  - — Supports both Standard and Fast-Page-Mode Accesses
- ■ Multiplexed Address Bus
  - — RAS# and CAS# Control Inputs
- No-Glue Interface to Many Memory Controllers
- SmartVoltage Technology
  - — User-Selectable 3.3V or 5V V_{CC}
  - — User-Selectable 5V or 12V V_{PP}
- 0.33 MB/sec Write Transfer Rate
- x16 Architecture
- 56-Lead TSOP Type I Package
- Backwards-Compatible with 28F008SA Command Set
- 2 µA Typical Deep Power-Down Current
- 1 mA Typical I_{CC} Active Current in Static Mode
- 32 Separately-Erasable/Lockable 64-Kbyte Blocks
- 1 Million Erase Cycles per Block
- State-of-the-Art 0.6 µm ETOX™ IV Flash Technology

Intel's 28F016XD 16-Mbit flash memory is a revolutionary architecture which is the ideal choice for designing truly revolutionary high-performance products. Combining its DRAM-like read performance and interface with the intrinsic nonvolatility of flash memory, the 28F016XD eliminates the traditional redundant memory paradigm of shadowing code from a slow nonvolatile storage source to a faster execution memory, such as DRAM, for improved system performance. The innovative capabilities of the 28F016XD enable the design of direct-execute code and mass storage data/file flash memory systems.

The 28F016XD's DRAM-like interface with a multiplexed address bus, flexible V_{CC} and V_{PP} voltages, power saving features, extended cycling, fast program and read performance, symmetrically-blocked architecture, and selective block locking provide a highly flexible memory component suitable for resident flash component arrays on the system board or SIMMs. The DRAM-like interface with RAS# and CAS# control inputs allows for easy migration to flash memory in existing DRAM-based systems. The 28F016XD's dual read voltage allows the same component to operate at either 3.3V or 5.0V V_{CC}. Programming voltage at 5.0V V_{PP} minimizes external circuitry in minimal-chip, space critical designs, while the 12.0V V_{PP} option maximizes program/erase performance. The x16 architecture allows optimization of the memory-to-processor interface. Its high read performance combined with flexible block locking enable both storage and execution of operating systems/application software and fast access to large data tables. The 28F016XD is manufactured on Intel's 0.6 µm ETOX IV process technology.

The datasheet is intended to give an overview of the chip feature-set and of the operating AC/DC specifications. *The 16-Mbit Flash Product Family User's Manual* provides complete descriptions of the user modes, system interface examples and detailed descriptions of all principles of operation. It also contains the full list of software algorithm flowcharts, and a brief section on compatibility with the Intel 28F008SA.

Significant 28F016XD feature revisions occurred between datasheet revisions 290533-001 and 290533-002. These revisions center around removal of the following features:

- All page buffer operations (read, write, programming, Upload Device Information)
- Command queuing
- Software Sleep and Abort
- Erase all Unlocked Blocks
- Device Configuration command

In addition, a significant 28F016XD change occurred between datasheet revisions 290532-002 and 290532-003. This change centers around the addition of a 3/5# pin to the device's pinout configuration. Figure 2 shows the 3/5# pin assignment for the TSOP Type 1 package.

Intel recommends that all customers obtain the latest revisions of 28F016XD documentation.

1.1 Product Overview

The 28F016XD is a high-performance, 16-Mbit (16,777,216-bit) block erasable, nonvolatile random access memory, organized as 1 Mword x 16. The 28F016XD includes thirty-two 32-KW (32,768 word) blocks. A chip memory map is shown in Figure 3.

- No-Glue Interface to Memory Controllers
- Improved Word Program Performance
- SmartVoltage Technology
  - — Selectable 3.3V or 5.0V V_{CC}
  - — Selectable 5.0V or 12.0V V_{PP}
- Block Program/Erase Protection

The 28F016XD's multiplexed address bus with RAS# and CAS# inputs allows for a "No Glue" interface to many existing in-system memory controllers. As such, 28F016XD-based SIMMs (72-pin JEDEC Standard) offer attractive advantages over their DRAM counterparts in many applications. For more information on 28F016XD-based SIMM designs, see the application note referenced at the end of this datasheet.

The 28F016XD incorporates SmartVoltage technology, providing V_{CC} operation at both 3.3V and 5.0V and program and erase capability at V_{PP} = 12.0V or 5.0V. Operating at V_{CC} = 3.3V, the 28F016XD consumes less than 60% of the power consumption at 5.0V V_{CC}, while 5.0V V_{CC} provides the highest read performance capability. V_{PP} = 5.0V operation eliminates the need for a separate 12.0V converter, while V_{PP} = 12.0V maximizes program/erase performance. In addition to the flexible program and erase voltages, the dedicated V_{PP} gives complete code protection with $V_{PP} \leq V_{PPLK}$.

A 3/5# input pin configures the device's internal circuitry for optimal 3.3V or 5.0V read/program operation.

A Command User Interface (CUI) serves as the system interface between the microprocessor or microcontroller and the internal memory operation.

Internal Algorithm Automation allows word programs and block erase operations to be executed using a Two-Write command sequence to the CUI in the same way as the 28F008SA 8-Mbit FlashFile™ memory.

Software Locking of Memory Blocks is an added feature of the 28F016XD as compared to the 28F008SA. The 28F016XD provides selectable block locking to protect code or data such as direct-executable operating systems or application code. Each block has an associated nonvolatile lock-bit which determines the lock status of the block. In addition, the 28F016XD has a master Write Protect pin (WP#) which prevents any modifications to memory blocks whose lock-bits are set.

Writing of memory data is performed in word increments typically within 6 µs (12.0V V_{PP})—a 33% improvement over the 28F008SA. A block erase operation erases one of the 32 blocks in typically 0.6 sec (12.0V V_{PP}), independent of the other blocks, which is about a 65% improvement over the 28F008SA.

Each block can be written and erased a minimum of 100,000 cycles. Systems can achieve one million Block Erase Cycles by providing wear-leveling algorithms and graceful block retirement. These techniques have already been employed in many flash file systems and hard disk drive designs.

All operations are started by a sequence of Write commands to the device. Three types of Status Registers (described in detail later in this datasheet) and a RY/BY# output pin provide information on the progress of the requested operation.

The following Status Registers are used to provide device and WSM information to the user :

- A Compatible Status Register (CSR) which is 100% compatible with the 28F008SA FlashFile memory Status Register. The CSR, when used alone, provides a straightforward upgrade capability to the 28F016XD from a 28F008SA-based design.
- A Global Status Register (GSR) which also informs the system of overall Write State Machine (WSM) status.
- 32 Block Status Registers (BSRs) which provide block-specific status information such as the block lock-bit status.

The GSR and BSR memory maps are shown in Figure 4.

The 28F016XD incorporates an open drain RY/BY# output pin. This feature allows the user to OR-tie many RY/BY# pins together in a multiple memory configuration such as a Resident Flash Array.

The 28F016XD is specified for a maximum fast page mode cycle time of 65 ns ($t_{PC,R}$) at 5.0V operation (4.75V to 5.25V) over the commercial temperature range (0°C to +70°C). A corresponding maximum fast page mode cycle time of 75 ns at 3.3V (3.0V to 3.6V and 0°C to +70°C) is achieved for reduced power consumption applications.

The 28F016XD incorporates an Automatic Power Saving (APS) feature, which substantially reduces the active current when the device is in static mode of operation (addresses not switching). In APS mode, the typical I_{CC} current is 1 mA at 5.0V (3.0 mA at 3.3V).

A deep power-down mode of operation is invoked when the RP# (called PWD# on the 28F008SA) pin transitions low. This mode brings the device power consumption to less than 2.0 µA, typically, and provides additional write protection by acting as a device reset pin during power transitions. A reset time of 300 ns (5.0V V_{CC} operation) is required from RP# switching high until dropping RAS#. In the deep power-down state, the WSM is reset (any current operation will abort) and the CSR, GSR and BSR registers are cleared.

A CMOS standby mode of operation is enabled when RAS# and CAS# transition high and RP# stays high with all input control pins at CMOS levels. In this mode, the device typically draws an I_{CC} standby current of 70 µA at 5.0V V_{CC}.

The 28F016XD is available in a 56-Lead, 1.2 mm thick, 14 mm x 20 mm TSOP Type I package. This form factor and pinout allow for very high board layout densities.

2.0 ADDITIONAL INFORMATION

Order Number	Document/Tool
297372	*16-Mbit Flash Product Family User's Manual*
292092	*AP-357 Power Supply Solutions for Flash Memory*
292123	*AP-374 Flash Memory Write Protection Techniques*
292126	*AP-377 16-Mbit Flash Product Family Software Drivers, 28F016SA/SV/XD/XS*
292131	*AP-384 Designing with the 28F016XD*
292163	*AP-610 Flash Memory In-System Code and Data Update Techniques*
292168	*AP-614 Adapting DRAM Based Designs for the 28F016XD*
292152	*AB-58 28F016XD-Based SIMM Designs*
292165	*AB-62 Compiled Code Optimizations for Flash Memories*
294016	*ER-33 ETOX™ Flash Memory Technology—Insight to Intel's Fourth Generation Process Innovation*
297508	FLASHBuilder Utility
Contact Intel/Distribution Sales Office	28F016XD Benchmark Utility
Contact Intel/Distribution Sales Office	Flash Cycling Utility
Contact Intel/Distribution Sales Office	28F016XD iBIS Models
Contact Intel/Distribution Sales Office	28F016XD VHDL/Verilog Models
Contact Intel/Distribution Sales Office	28F016XD TimingDesigner* Library Files
Contact Intel/Distribution Sales Office	28F016XD Orcad and ViewLogic Schematic Symbols

This complete document is available from Intel's World Wide Website and/or U.S. Literature Center:
World Wide Website: http://www.Intel.com
U. S. Literature Center: 800-548-4725
In other geographies, please contact your local sales office.

- **30.8 MB/sec Burst Write Transfer Rate**
- **0.48 MB/sec Sustainable Write Transfer Rate**
- **Configurable x8 or x16 Operation**
- **56-Lead TSOP and SSOP Type I Packages**
- **Command Superset of the Intel 28F008SA**
  - **Page Buffer Program**
- **2 µA Typical Deep Power-Down**
- **32 Independently Lockable Blocks**
- **State-of-the-Art 0.6 µm ETOX™ IV Flash Technology**

Intel's 28F016SV 16-Mbit FlashFile™ memory is a revolutionary architecture which is the ideal choice for designing embedded direct-execute code and mass storage data/file flash memory systems. With innovative capabilities, low-power operation, user-selectable V_{PP} voltage and high read/program performance, the 28F016SV enables the design of truly mobile, high-performance personal computing and communications products.

The 28F016SV is the highest density, highest performance nonvolatile read/program solution for solid-state storage applications. Its symmetrically-blocked architecture (100% compatible with the 28F008SA 8-Mbit and 28F016SA 16-Mbit FlashFile memories), extended cycling, flexible V_{CC} and V_{PP} voltage (SmartVoltage technology), fast program and read performance and selective block locking, provide a highly-flexible memory component suitable for Resident Flash Arrays, high-density memory cards and PCMCIA-ATA flash drives. The 28F016SV's dual read voltage enables the design of memory cards which can be read/written in 3.3V and 5V systems interchangeably. Its x8/x16 architecture allows optimization of the memory-to-processor interface. The flexible block locking option enables bundling of executable application software in a Resident Flash Array or memory card. The 28F016SV is manufactured on Intel's 0.6 µm ETOX IV process technology.

1.0 INTRODUCTION

The documentation of the Intel 28F016SV memory device includes this datasheet, a detailed user's manual, and a number of application notes and design tools, all of which are referenced in Appendix B.

The datasheet is intended to give an overview of the chip feature-set and of the operating AC/DC specifications. The *16-Mbit Flash Product Family User's Manual* provides complete descriptions of the user modes, system interface examples and detailed descriptions of all principles of operation. It also contains the full list of software algorithm flowcharts, and a brief section on compatibility with the Intel 28F008SA.

A significant 28F016SV change occurred between datasheet revisions 290528-003 and 290528-004. This change centers around the addition of a 3/5# pin to the device's pinout configuration. Figures 2 and 3 show the 3/5# pin assignment for TSOP and SSOP Type 1 packages. Intel recommends that all customers obtain the latest revisions of 28F016SV documentation.

1.1 Enhanced Features

The 28F016SV is backwards compatible with the 28F016SA and offers the following enhancements:

- SmartVoltage Technology
  - — Selectable 5V or 12V V_{PP}
- V_{PP} Level Bit in Block Status Register
- Additional RY/BY# Configuration
  - — Pulse-On-Program/Erase
- Additional Upload Device Information Command Feedback
  - — Device Proliferation Code
  - — Device Configuration Code

1.2 Product Overview

The 28F016SV is a high-performance, 16-Mbit (16,777,216-bit) block erasable, nonvolatile random access memory, organized as either 1 Mword x 16 or 2 Mbyte x 8. The 28F016SV includes thirty-two 64-KB (65,536 byte) blocks or thirty-two 32-KW (32,768 word) blocks. A chip memory map is shown in Figure 4.

The implementation of a new architecture, with many enhanced features, will improve the device operating characteristics and result in greater product reliability and ease-of-use.

The 28F016SV incorporates SmartVoltage technology, providing V_{CC} operation at both 3.3V and 5V and program and erase capability at V_{PP} = 12V or 5V. Operating at V_{CC} = 3.3V, the 28F016SV consumes approximately one half the power consumption at 5V V_{CC}, while 5V V_{CC} provides the highest read performance capability. V_{PP} = 5V operation eliminates the need for a separate 12V converter, while V_{PP} = 12V maximizes program/erase performance. In addition to the flexible program and erase voltages, the dedicated V_{PP} gives complete code protection with $V_{PP} \leq V_{PPLK}$.

A 3/5# input pin configures the device's internal circuitry for optimal 3.3V or 5V read/program operation.

A Command User Interface (CUI) serves as the system interface between the microprocessor or microcontroller and the internal memory operation.

Internal Algorithm Automation allows byte/word programs and block erase operations to be executed using a Two-Program command sequence to the CUI in the same way as the 28F008SA 8-Mbit FlashFile™ memory.

A super-set of commands has been added to the basic 28F008SA command-set to achieve higher program performance and provide additional capabilities. These new commands and features include:

- Page Buffer Programs to Flash
- Command Queuing Capability
- Automatic Data Programs during Erase
- Software Locking of Memory Blocks
- Two-Byte Successive Programs in 8-bit Systems
- Erase All Unlocked Blocks

Writing of memory data is performed in either byte or word increments typically within 6 µs (12V V_{PP})—a 33% improvement over the 28F008SA. A block erase operation erases one of the 32 blocks in typically 0.6 sec (12V V_{PP}), independent of the other blocks, which is about a 65% improvement over the 28F008SA.

Each block can be written and erased a minimum of 100,000 cycles. Systems can achieve one million Block Erase Cycles by providing wear-leveling algorithms and graceful block retirement. These techniques have already been employed in many flash file systems and hard disk drive designs.

The 28F016SV incorporates two Page Buffers of 256 bytes (128 words) each to allow page data programs. This feature can improve a system program performance by up to 4.8 times over previous flash memory devices, which have no Page Buffers.

All operations are started by a sequence of Program commands to the device. Three Status Registers (described in detail later in this datasheet) and a RY/BY# output pin provide information on the progress of the requested operation.

While the 28F008SA requires an operation to complete before the next operation can be requested, the 28F016SV allows queuing of the next operation while the memory executes the current operation. This eliminates system overhead when writing several bytes in a row to the array or erasing several blocks at the same time. The 28F016SV can also perform program operations to one block of memory while performing erase of another block.

The 28F016SV provides selectable block locking to protect code or data such as Device Drivers, PCMCIA card information, ROM-Executable O/S or Application Code. Each block has an associated nonvolatile lock-bit which determines the lock status of the block. In addition, the 28F016SV has a master Write Protect pin (WP#) which prevents any modifications to memory blocks whose lock-bits are set.

The 28F016SV contains three types of Status Registers to accomplish various functions:

- A Compatible Status Register (CSR) which is 100% compatible with the 28F008SA FlashFile memory Status Register. The CSR, when used alone, provides a straightforward upgrade capability to the 28F016SV from a 28F008SA-based design.
- A Global Status Register (GSR) which informs the system of command Queue status, Page Buffer status, and overall Write State Machine (WSM) status.
- 32 Block Status Registers (BSRs) which provide block-specific status information such as the block lock-bit status.

The GSR and BSR memory maps for byte-wide and word-wide modes are shown in Figures 5 and 6.

The 28F016SV incorporates an open drain RY/BY# output pin. This feature allows the user to OR-tie many RY/BY# pins together in a multiple memory configuration such as a Resident Flash Array.

Other configurations of the RY/BY# pin are enabled via special CUI commands and are described in detail in the *16-Mbit Flash Product Family User's Manual*.

The 28F016SV's enhanced Upload Device Information command provides access to additional information that the 28F016SA previously did not offer. This command uploads the Device Revision Number, Device Proliferation Code and Device Configuration Code to the page buffer. The Device Proliferation Code for the 28F016SV is 01H, and the Device Configuration Code identifies the current RY/BY# configuration. Section 4.4 documents the exact page buffer address locations for all uploaded information. A subsequent Page Buffer Swap and Page Buffer Read command sequence is necessary to read the correct device information.

The 28F016SV also incorporates a dual chip-enable function with two input pins, CE_0# and CE_1#. These pins have exactly the same functionality as the regular chip-enable pin, CE#, on the 28F008SA. For minimum chip designs, CE_1# may be tied to ground and system logic may use CE_0# as the chip enable input. The 28F016SV uses the logical combination of these two signals to enable or disable the entire chip. Both CE_0# and CE_1# must be active low to enable the device. If either one becomes inactive, the chip will be disabled. This feature, along with the open drain RY/BY# pin, allows the system designer to reduce the number of control pins used in a large array of 16-Mbit devices.

The BYTE# pin allows either x8 or x16 read/programs to the 28F016SV. BYTE# at logic low selects 8-bit mode with address A_0 selecting between the low byte and high byte. On the other hand, BYTE# at logic high enables 16-bit operation with address A_1 becoming the lowest order address and address A_0 is not used (don't care). A device block diagram is shown in Figure 1.

The 28F016SV is specified for a maximum access time of 65 ns (t_{ACC}) at 5V operation (4.75V to 5.25V) over the commercial temperature range (0°C to +70°C). A corresponding maximum access time of 75 ns at 3.3V (3.0V to 3.6V and 0°C to +70°C) is achieved for reduced power consumption applications.

The 28F016SV incorporates an Automatic Power Saving (APS) feature, which substantially reduces the active current when the device is in static mode of operation (addresses not switching). In APS mode, the typical I_{CC} current is 1 mA at 5V (3.0 mA at 3.3V).

A deep power-down mode of operation is invoked when the RP# (called PWD# on the 28F008SA) pin transitions low. This mode brings the device power consumption to less than 2.0 µA, typically, and provides additional program protection by acting as a device reset pin during power transitions. A reset time of 400 ns (5V V_{CC} operation) is required from RP# switching high until outputs are again valid. In the Deep Power-Down state, the WSM is reset (any current operation will abort) and the CSR, GSR and BSR registers are cleared.

A CMOS standby mode of operation is enabled when either CE_0# or CE_1# transitions high and RP# stays high with all input control pins at CMOS levels. In this mode, the device typically draws an I_{CC} standby current of 70 µA at 5V V_{CC}.

The 28F016SV will be available in 56-lead, 1.2 mm thick, 14 mm x 20 mm TSOP and 56-lead, 1.8 mm thick, 16 mm x 23.7 SSOP Type I packages. The form factor and pinout of these two packages allow for very high board layout densities.

2.0 ADDITIONAL INFORMATION

Order Number	Document/Tool
297372	*16-Mbit Flash Product Family User's Manual*
290429	*28F008SA Datasheet*
290490	*DD28F032SA 32-Mbit (2 bit x 16, 4 Mbit x 8) FlashFile™ Memory Datasheet)*
292092	*AP-357 Power Supply Solutions for Flash Memory*
292123	*AP-374 Flash Memory Write Protection Techniques*
292126	*AP-377 16-Mbit Flash Product Family Software Drivers, 28F016SA/28F016SV/28F016XS/28F016XD*
292144	*AP-393 28F016SV Compatibility with 28F016SA*
292159	*AP-607 Multi-Site Layout Planning with Intel's FlashFile™ Components, Including ROM Capability*
292163	*AP-610 Flash Memory In-System Code and Data Update Techniques*
292165	*AB-62 Compiled Code Optimizations for Flash Memories*
294016	*ER-33 ETOX™ Flash Memory Technology—Insight to Intel's Fourth Generation Process Innovation*
297508	FLASHBuilder Utility
Contact Intel/Distribution Sales Office	Flash Cycling Utility
Contact Intel/Distribution Sales Office	28F016SV iBIS Model
Contact Intel/Distribution Sales Office	28F016SV VHDL
Contact Intel/Distribution Sales Office	28F016SV Timing Designer Library Files
Contact Intel/Distribution Sales Office	28F016SV Orcad and ViewLogic Schematic Symbols

This complete document is available from Intel's World Wide Website and/or U.S. Literature Center:
World Wide Website: http://www.Intel.com
U. S. Literature Center: 800-548-4725
In other geographies, please contact your local sales office.

292092	AP-357 Power Supply Solutions for Flash Memory
292123	AP-374 Flash Memory Write Protection Techniques
292126	AP-377 16-Mbit Flash Product Family Software Drivers: 28F016SA/28F016SV/28F016X/28F016XD
292144	AP-393 28F016SV Compatibility with 28F016SA
292159	AP-807 Multi-Site Layout Planning with Intel's FlashFile™ Components, Including ROM Capability
292163	AP-810 Flash Memory In-System Code and Data Update Techniques
292165	AB-62 Compiled Code Optimizations for Flash Memories
294016	EP-33 ETOX™ Flash Memory Technology—Insight to Intel's Fourth Generation Process Innovation
297508	FLASHBuilder Utility
Contact Intel/Distribution Sales Office	Flash Cycling Utility
Contact Intel/Distribution Sales Office	28F016SV IBIS Model
Contact Intel/Distribution Sales Office	28F016SV VHDL
Contact Intel/Distribution Sales Office	28F016SV Timing Designer Library Files
Contact Intel/Distribution Sales Office	28F016SV Orcad and Viewlogic Schematic Symbols

This complete document is available from Intel's World Wide Website and/or U.S. Literature Center:
World Wide Website: http://www.intel.com
U. S. Literature Center: 800-548-4725
In other geographies, please contact your local sales office.

intel®

AB-60

APPLICATION BRIEF

2/4/8-Mbit SmartVoltage Boot Block Flash Memory Family Overview

September 1995

Order Number: 292154-002

intel

AB-60

APPLICATION
BRIEF

2/4/8-Mbit SmartVoltage Boot Block Flash Memory Family Overview

September 1995

1.0 INTRODUCTION

This document includes a feature overview, pinouts, and memory maps for Intel's SmartVoltage boot block family, including 2/4/8-Mbit densities. These products offer feature and function compatibility with the boot block architecture, along with the SmartVoltage technology (SVT) outlined below. Follow the design steps in Section 5.0 to upgrade 12V V_{PP} designs to SVT.

2.0 BOOT BLOCK ARCHITECTURE

Intel's boot block architecture products offer the familiar features that optimize it for updateable firmware storage. These features include:

- Hardware-lockable boot block for secure kernel code storage
- Parameter blocks for parameter storage
- Main blocks for modular code updates, facilitating updateable firmware
- x8 or x16 user-selectable I/O operation
- RP# for reset and write protection
- PSOP and TSOP packages

Intel has integrated its SmartVoltage technology into the boot block family in order to increase the voltage flexibility of these components.

3.0 PINOUT COMPATIBLE DENSITY UPGRADES

In addition, Intel is providing density upgrades with pinout compatibility for the 2-Mbit, 4-Mbit, and 8-Mbit densities. The pinouts in Figures 2, 3, and 4 illustrate these compatible upgrade paths.

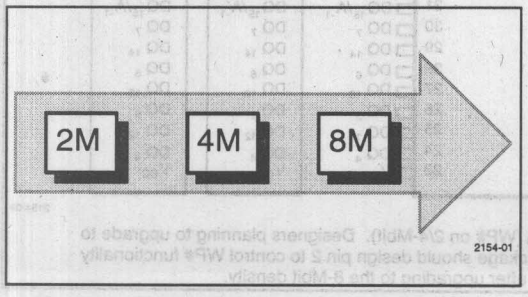


Figure 1. The SmartVoltage Technology Boot Block Line Features a Pinout-Compatible Upgrade Path

4.0 NEW SmartVoltage TECHNOLOGY FEATURES

SmartVoltage offers the following new features:

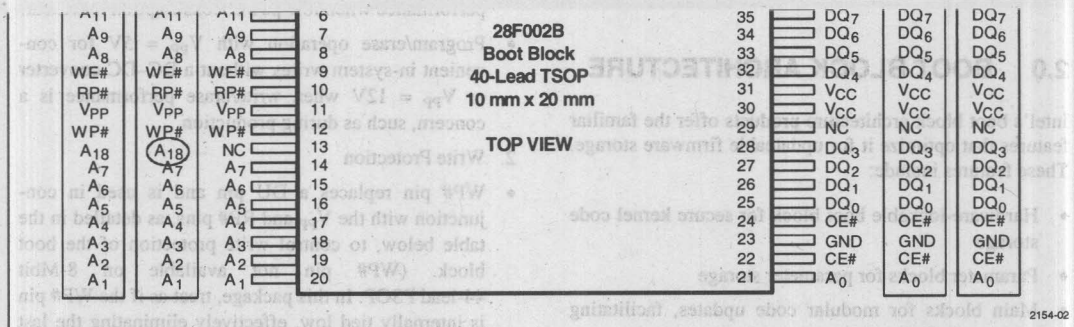
1. Voltage Flexibility
   - V_{CC} = 2.7–3.6V, 3.3V ±0.3V or 5V ±10% with enhanced circuits to optimize low-voltage performance when low power consumption is critical.
   - Program/erase operation with V_{PP} = 5V for convenient in-system writes without a DC–DC converter or V_{PP} = 12V when write/erase performance is a concern, such as during production.
2. Write Protection
   - WP# pin replaces a DU pin and is used in conjunction with the V_{PP} and RP# pins, as detailed in the table below, to control write protection of the boot block. (WP# pin not available on 8-Mbit 44-lead PSOP. In this package, treat as if the WP# pin is internally tied low, effectively eliminating the last row of the table below.)

V_{PP}	RP#	WP#	Write Protection
V_{IL}	X	X	All Blocks Locked
≥ V_{PPLK}	V_{IL}	X	All Blocks Locked (Reset)
≥ V_{PPLK}	V_{HH}	X	All Blocks Unlocked
≥ V_{PPLK}	V_{IH}	V_{IL}	Boot Block Locked
≥ V_{PPLK}	V_{IH}	V_{IH}	All Blocks Unlocked

5.0 UPGRADING FROM 12V TO SVT

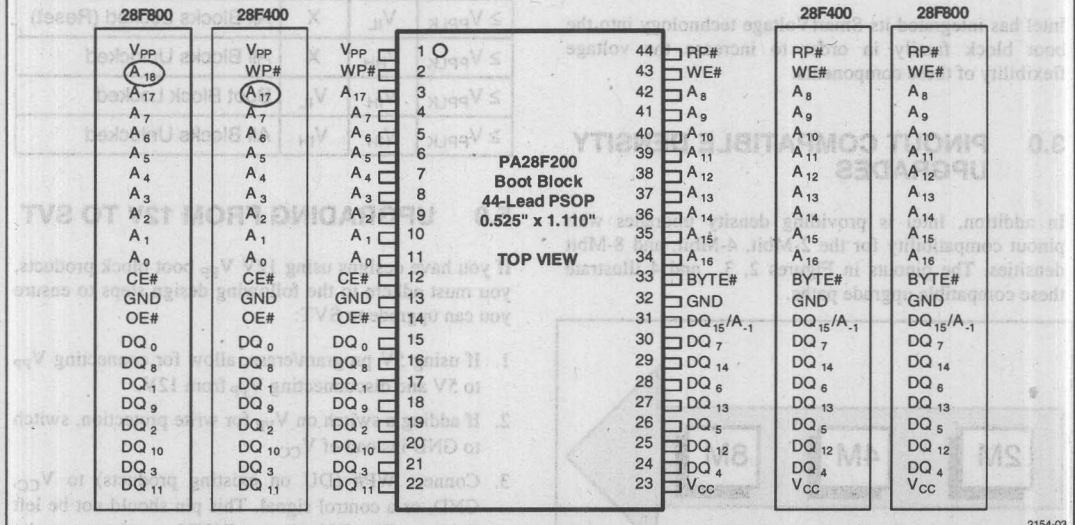
If you have designs using 12V V_{PP} boot block products, you must adhere to the following design steps to ensure you can upgrade to SVT:

1. If using 5V program/erase, allow for connecting V_{PP} to 5V and disconnecting V_{PP} from 12V.
2. If adding a switch on V_{PP} for write protection, switch to GND instead of V_{CC}.
3. Connect WP# (DU on existing products) to V_{CC}, GND, or a control signal. This pin should not be left floating. The DU pin on BX/BL products can be driven to a logic-level in order to provide upgrade compatibility.



NOTE: The 28F008B pinout shown is for the 8-Mbit boot block, and not the 28F008SA FlashFile™ memory.

Figure 2. The 40-Lead TSOP Offers the Smallest Form Factor for Space-Constrained Applications



NOTE: For the 8-Mbit device, pin 2 has been changed to A_{18} (WP# on 2/4-Mbit). Designers planning to upgrade to the 8-Mbit density from the 2/4-Mbit density in this package should design pin 2 to control WP# functionality at the 2/4-Mbit level and allow for pin 2 to control A_{18} after upgrading to the 8-Mbit density.

Figure 3. The 44-Lead PSOP Offers a Convenient Upgrade from JEDEC ROM Standards

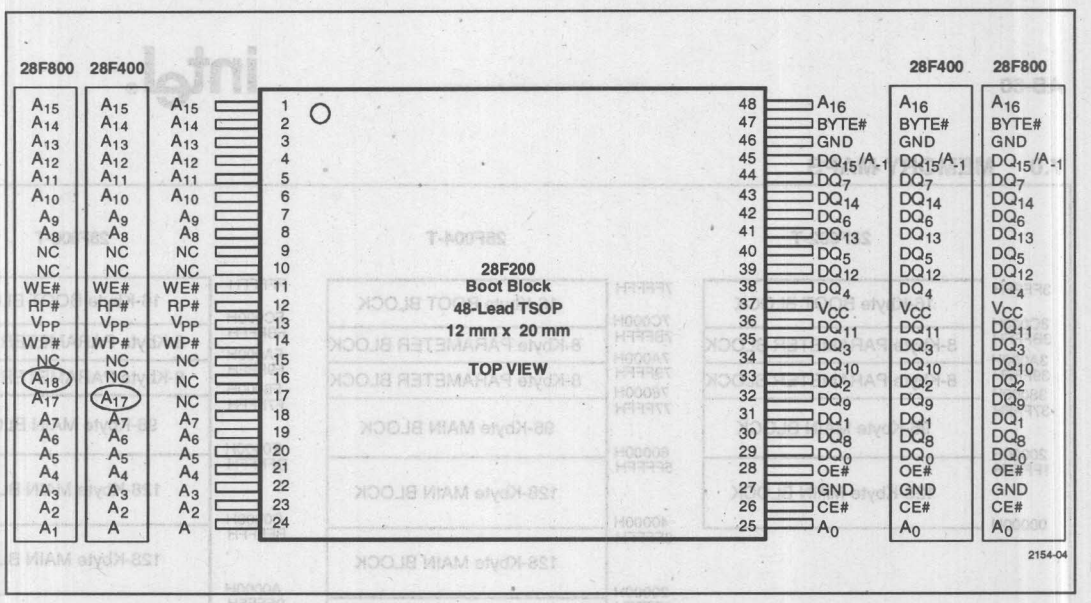


Figure 4. The 48-Lead TSOP Offers the Smallest Form Factor for x16 Operation

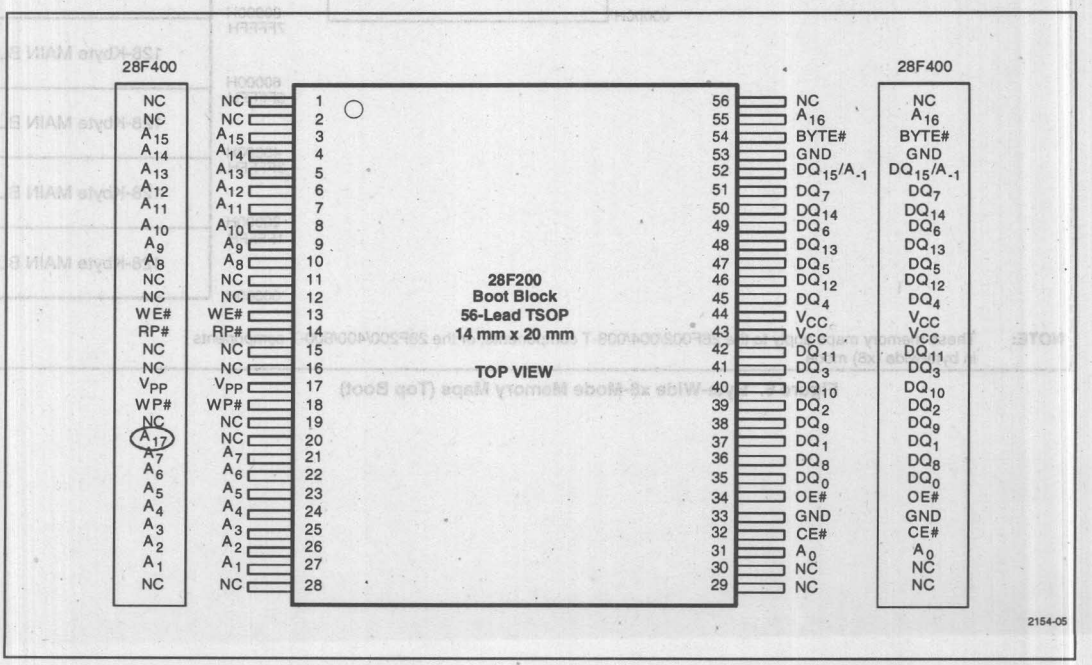
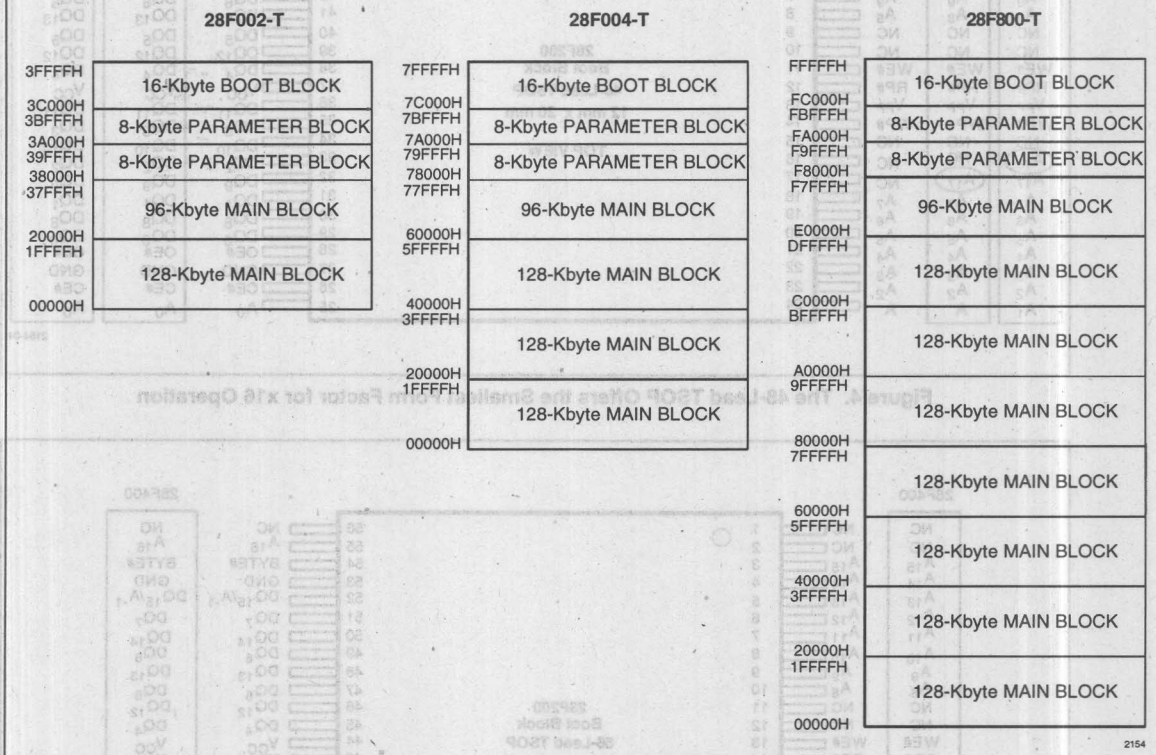


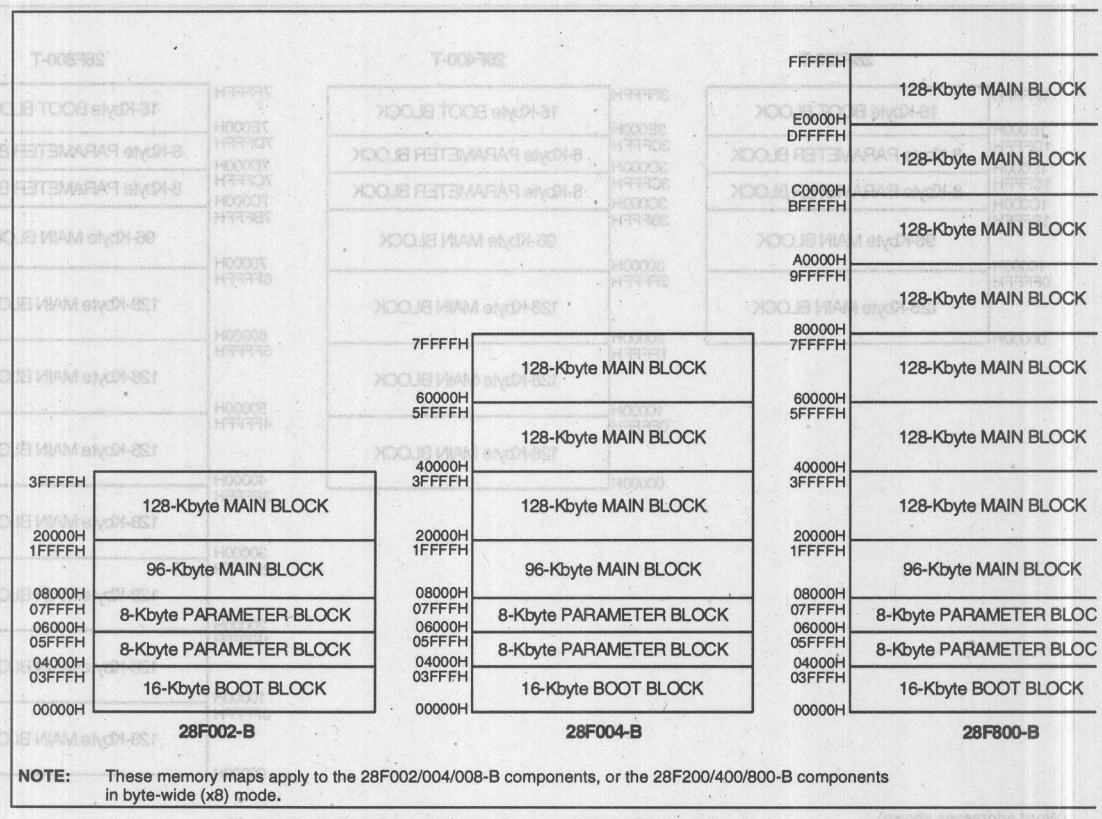
Figure 5. The 56-Lead TSOP Offers Compatibility between 2 and 4 Mbits

7.0 MEMORY MAPS



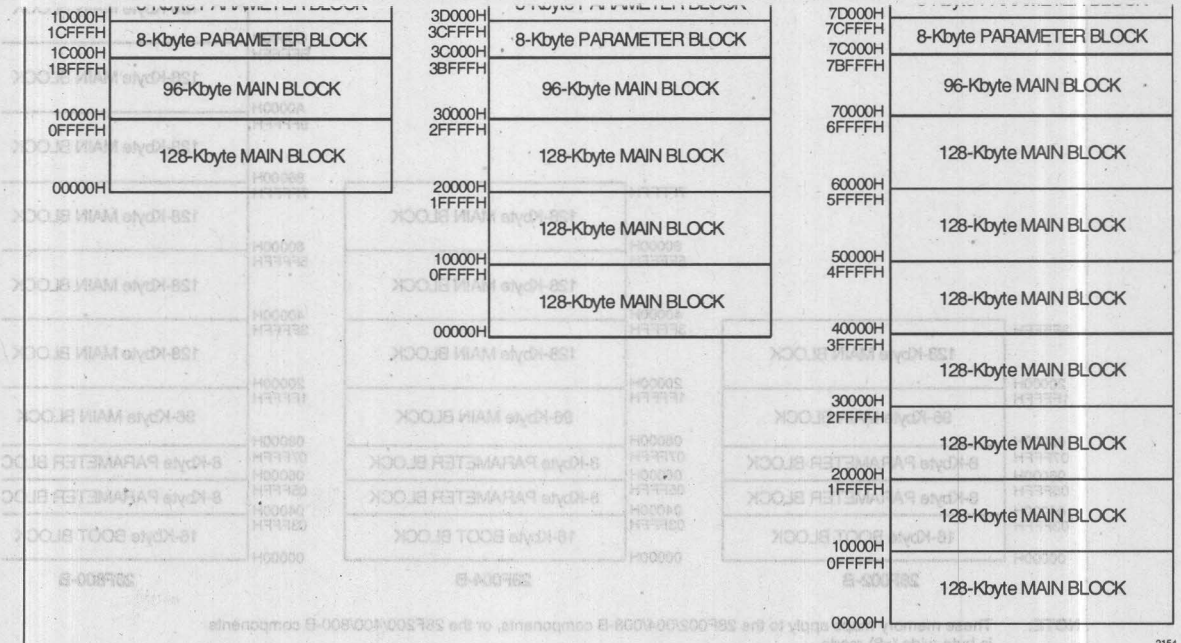
NOTE: These memory maps apply to the 28F002/004/008-T components, or the 28F200/400/800-T components in byte-wide (x8) mode.

Figure 6. Byte-Wide x8-Mode Memory Maps (Top Boot)



NOTE: These memory maps apply to the 28F002/004/008-B components, or the 28F200/400/800-B components in byte-wide (x8) mode.

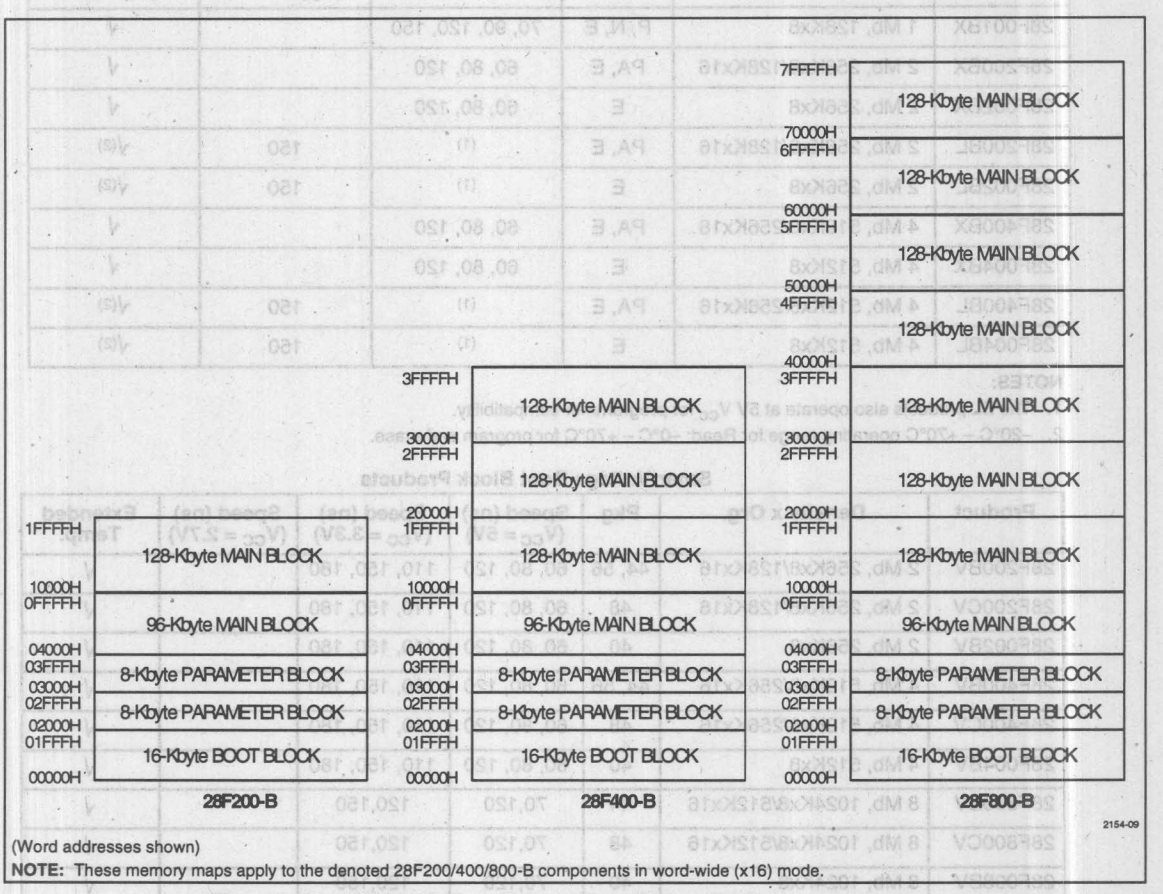
Figure 7. Byte-Wide x8-Mode Memory Maps (Bottom Boot)



(Word addresses shown)

NOTE: These memory maps apply to the denoted 28F200/400/800-T components in word-wide (x16) mode.

Figure 8. Word-Wide x16-Mode Memory Maps (Top Boot)



(Word addresses shown)

NOTE: These memory maps apply to the denoted 28F200/400/800-B components in word-wide (x16) mode.

Figure 9. Word-Wide x16-Mode Memory Maps (Bottom Boot)

8.0 PRODUCT OFFERINGS

12V Program/Erase Boot Block Products

Product	Density x Org.	Pkg.	Speed (ns) (V_{CC} = 5V)	Speed (ns) (V_{CC} = 3.3V)	Extended Temperature
28F001BX	1 Mb, 128Kx8	P, N, E	70, 90, 120, 150		√
28F200BX	2 Mb, 256Kx8/128Kx16	PA, E	60, 80, 120		√
28F002BX	2 Mb, 256Kx8	E	60, 80, 120		√
28F200BL	2 Mb, 256Kx8/128Kx16	PA, E	(1)	150	√(2)
28F002BL	2 Mb, 256Kx8	E	(1)	150	√(2)
28F400BX	4 Mb, 512Kx8/256Kx16	PA, E	60, 80, 120		√
28F004BX	4 Mb, 512Kx8	E	60, 80, 120		√
28F400BL	4 Mb, 512Kx8/256Kx16	PA, E	(1)	150	√(2)
28F004BL	4 Mb, 512Kx8	E	(1)	150	√(2)

NOTES:
1. The BL products also operate at 5V V_{CC} for programmer compatibility.
2. –20°C – +70°C operating range for Read; –0°C – +70°C for program and erase.

SmartVoltage Boot Block Products

Product	Density x Org.	Pkg	Speed (ns) (V_{CC} = 5V)	Speed (ns) (V_{CC} = 3.3V)	Speed (ns) (V_{CC} = 2.7V)	Extended Temp.
28F200BV	2 Mb, 256Kx8/128Kx16	44, 56	60, 80, 120	110, 150, 180		√
28F200CV	2 Mb, 256Kx8/128Kx16	48	60, 80, 120	110, 150, 180		√
28F002BV	2 Mb, 256Kx8	40	60, 80, 120	110, 150, 180		√
28F400BV	4 Mb, 512Kx8/256Kx16	44, 56	60, 80, 120	110, 150, 180		√
28F400CV	4 Mb, 512Kx8/256Kx16	48	60, 80, 120	110, 150, 180		√
28F004BV	4 Mb, 512Kx8	40	60, 80, 120	110, 150, 180		√
28F800BV	8 Mb, 1024Kx8/512Kx16	44	70,120	120,150		√
28F800CV	8 Mb, 1024Kx8/512Kx16	48	70,120	120,150		√
28F008BV	8 Mb, 1024Kx8	40	70,120	120,150		√
28F800CE	8 Mb, 1024Kx8/512Kx16	48	90		120	√
28F008BE	8 Mb, 1024Kx8/512Kx16	40	90		120	√

NOTES:
1. BV products also operate at V_{CC} = 5V for high-performance.
2. BE/CE products operate over the V_{CC} ranges 2.7–3.6V and 5V ±10%.

9.0 ORDERING INFORMATION

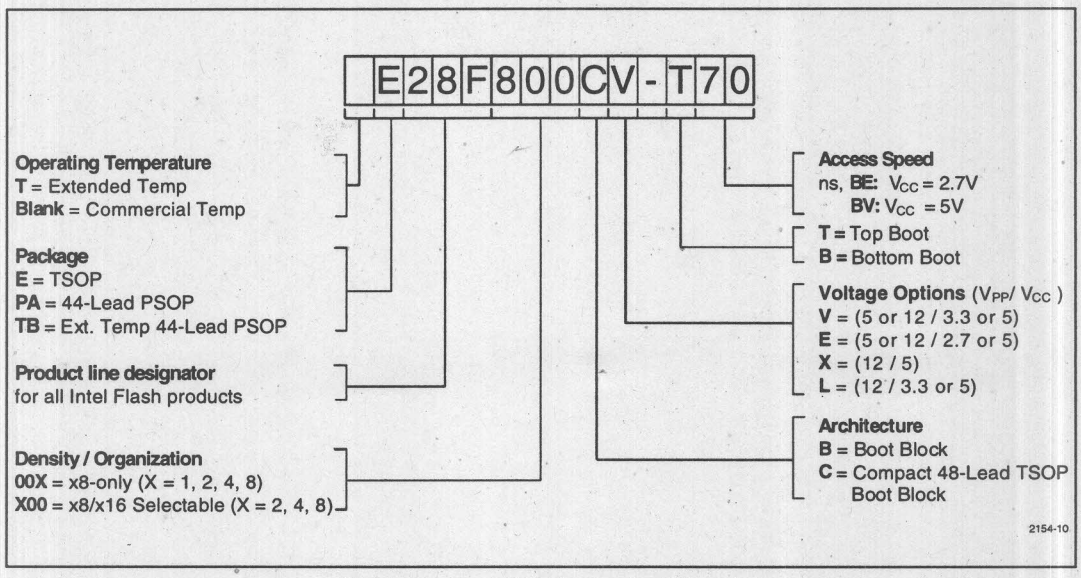


Figure 10. Decoding the Product Names

10.0 ADDITIONAL INFORMATION

10.1 Revision History

Number	Description
-001	Original Version
-002	Text updated. DU and WP# pin usage clarified. Note and "B" suffixes added to Figure 2. SmartVoltage Boot Block Products table: Speeds added to 28F800 and 28F008, 2.7V products added. Figure 10 updated.

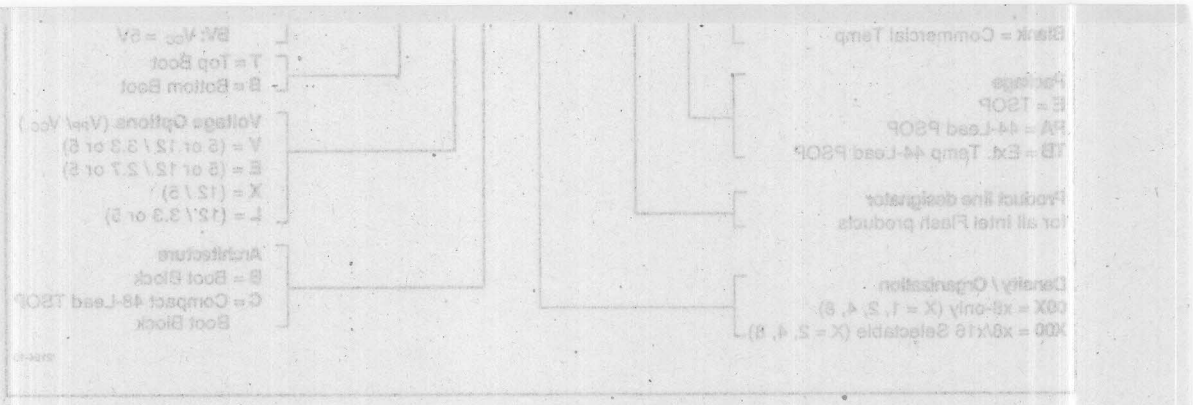


Figure 10. Decoding the Product Names

10.0 ADDITIONAL INFORMATION

10.1 Revision History

Number	Description
-001	Original Version
-002	Text updated. DU and WP# pin usage clarified. Note and "B" suffixes added to Figure 2. SmartVoltage Boot Block Products table: Speeds added to 28F800 and 28F008, 2.7V products added. Figure 10 updated.

intel®

TECHNICAL PAPER

Interfacing the 28F016XS to the i960® Microprocessor Family

November 1995

Order Number: 297500-003

intel

TECHNICAL
PAPER

Interfacing the 28F016XS to the i960® Microprocessor Family

November 1995

Order Number 297500-003

1.0 INTRODUCTION

This technical paper describes several designs interfacing the high-performance 28F016XS flash memory to the i960® microprocessor family. All designs are based on preliminary 28F016XS specifications. Please contact your Intel or distribution sales office for up-to-date specifications before finalizing any design.

The 28F016XS is a 16-Mbit block erasable flash memory with a high-performance synchronous pipelined read interface. This optimized interface can sustain a high read transfer rate and makes the 28F016XS the ideal flash memory component when interfacing to a burst processor, such as the i960 microprocessor. The 28F016XS combines ROM-like nonvolatility, DRAM-like read performance and in-system updateability in one memory technology. These characteristics enable code execution directly from the 28F016XS, replacing the costly practice of shadowing code from HDD or ROM to DRAM for improved performance. With the 28F016XS, you can construct a high-performance, cost-effective system solution for use with a burst microprocessor.

The 28F016XS performs synchronous pipelined reads. Up to three accesses can be initiated before reading data output from the initial cycle. This synchronous pipelined structure makes it ideal for use with the i960 microprocessor. The 28F016XS takes full advantage of the processor's burst transfer mechanism, bringing significant system performance enhancements to an i960 microprocessor-based application. This technical paper describes processor-to-memory interfaces that exploit these capabilities to achieve maximum system performance. Figures 1 and 2 illustrate relative system performance enhancements that the 28F016XS brings to an i960 microprocessor-based environment, compared to other memory technologies. The benchmark parameters are documented in Appendix B.

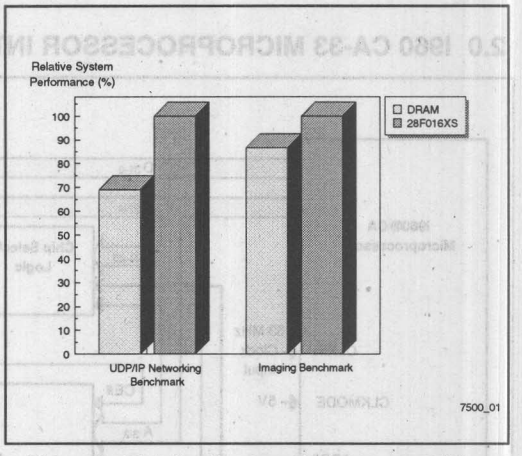


Figure 1. Relative System Performance of the 28F016XS Compared to Other Memory Technologies in an i960 KB-25 Microprocessor-Based Design

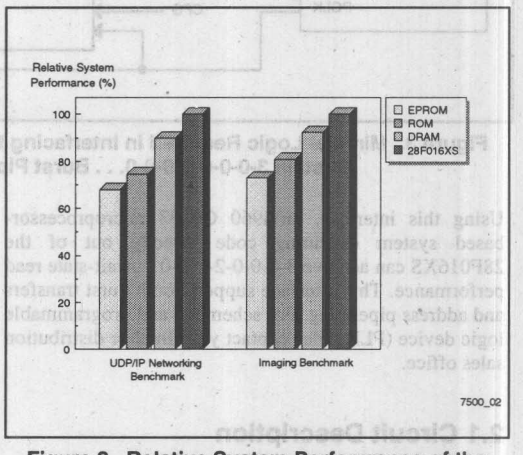


Figure 2. Relative System Performance of the 28F016XS Compared to Other Memory Technologies in an i960 CA-33 Processor-Based Design

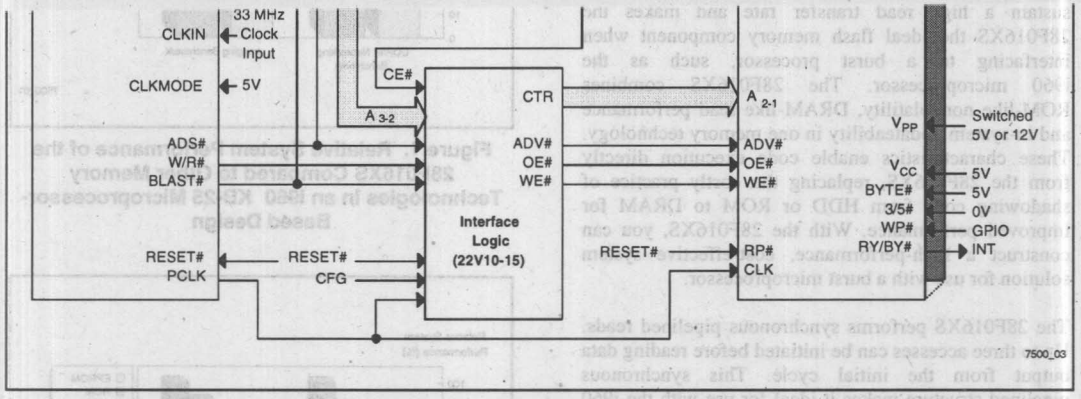


Figure 3. Minimal Logic Required in Interfacing the 28F016XS-15 to the i960 CA-33 Microprocessor to Sustain 3-0-0-0-2-0-0-0. . . Burst Pipelined Read Performance up to 33 MHz

Using this interface, an i960 CA-33 microprocessor-based system executing code directly out of the 28F016XS can achieve 3-0-0-0-2-0-0-0. . .wait-state read performance. This interface supports both burst transfers and address pipelining. For schematic and programmable logic device (PLD) files contact your Intel or distribution sales office.

2.1 Circuit Description

This section describes the 28F016XS-15 interface to the i960 CA-33 microprocessor. The block diagram for this design is illustrated in Figure 3.

Memory Configuration

This design uses two 28F016XS-15s, in x16 mode, to match the i960 CA microprocessor's 32-bit data bus, providing 4 Mbytes of flash memory. Signals A_{21-4} from the i960 CA microprocessor and CTR_{1-0} from the PLD select locations within the 28F016XS memory space, arranged as 1-Meg double words. A two-bit counter, implemented in the PLD, loads from the processor's lower addresses at the beginning of each memory cycle and generates subsequent burst addresses on outputs CTR_{1-0}. The counter feeds burst addresses to the 28F016XSs to prevent access stalling, while the devices wait for the processor to supply the next address.

Chip Select Logic

Chip select decode logic may use A_{31-22} to generate an active low chip select signal, CE#, for the 28F016XS-15 memory space and other system peripherals. The chip select drives CE_0# on each 28F016XS-15 and a control input to the PLD. The 28F016XS-15's CE_1# input is grounded.

In support of address pipelining, the chip select logic latches CE#, holding it active throughout the duration of the memory access. This will prevent potential CE# problems caused by using combinatorial logic when utilizing the address pipelining capability of i960 CA microprocessor.

If address pipelining functionality is not implemented, simple combinatorial logic can be utilized in generating the system CE# for the 28F016XS memory space, and the chip select logic shown in Figure 3 does not examine BLAST# and ADS#. For many systems using the upper address bits in a linear selection scheme may provide a sufficient number of chip select signals, thus eliminating system chip select decode logic. (See Figure 4 for an example of using linear selection for chip selects.) When using a linear chip select scheme however, the software must avoid using addresses that may select more than one device, which could result in bus contention.

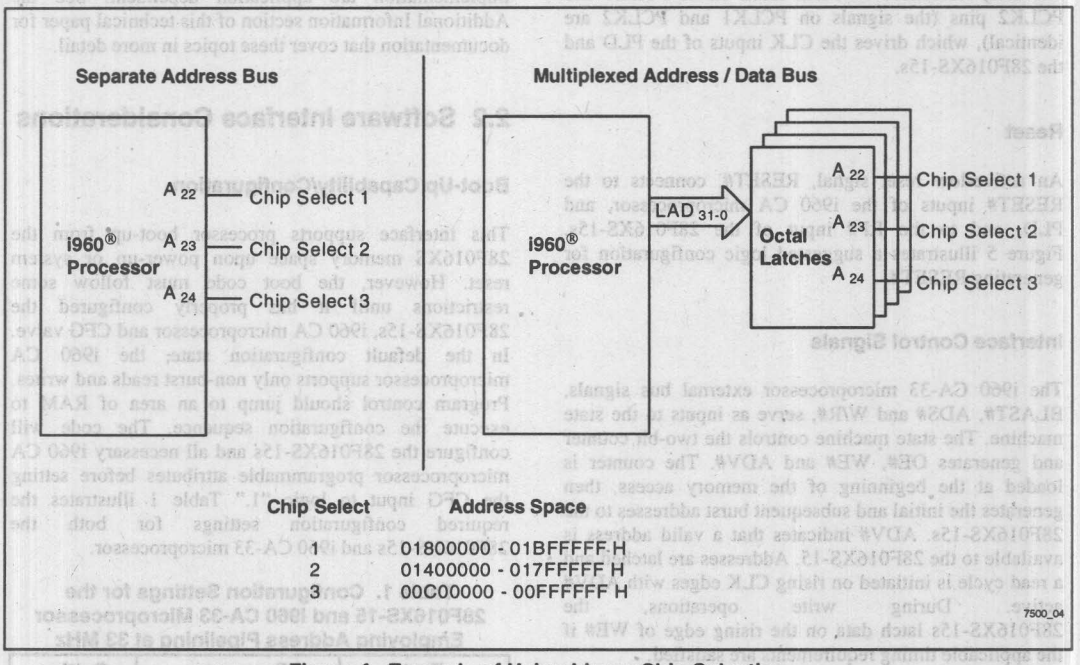


Figure 4. Example of Using Linear Chip Selection

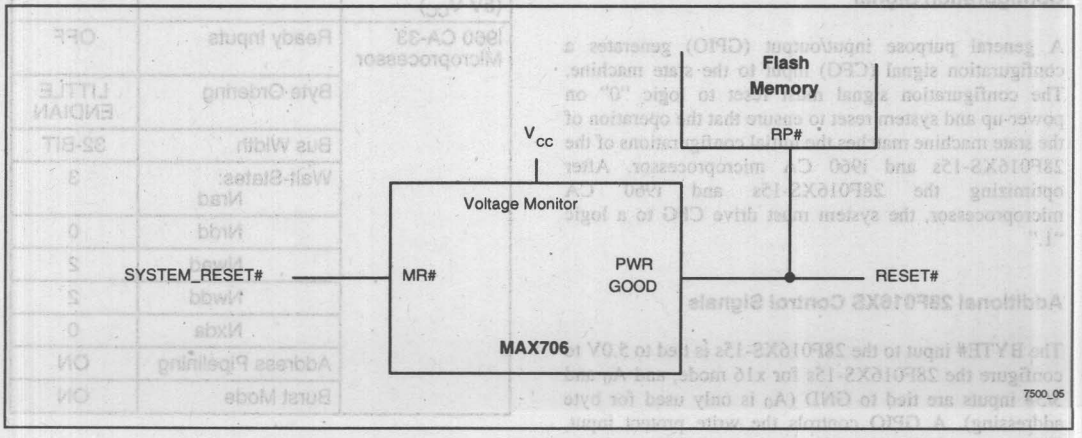


Figure 5. Example RESET Generation Circuitry

CLK Option

A 33 MHz clock signal drives the i960 CA microprocessor CLKIN input. Driving CLKMODE to a logic "1" configures the i960 CA microprocessor for a 1x CLK input. The i960 CA microprocessor outputs an internally-referenced 33 MHz clock on its PCLK1 and PCLK2 pins (the signals on PCLK1 and PCLK2 are identical), which drives the CLK inputs of the PLD and the 28F016XS-15s.

Reset

An active-low reset signal, RESET#, connects to the RESET# inputs of the i960 CA microprocessor, and PLD, and to the RP# input of the 28F016XS-15s. Figure 5 illustrates a suggested logic configuration for generating RESET#.

Interface Control Signals

The i960 CA-33 microprocessor external bus signals, BLAST#, ADS# and W/R#, serve as inputs to the state machine. The state machine controls the two-bit counter and generates OE#, WE# and ADV#. The counter is loaded at the beginning of the memory access, then generates the initial and subsequent burst addresses to the 28F016XS-15s. ADV# indicates that a valid address is available to the 28F016XS-15. Addresses are latched and a read cycle is initiated on rising CLK edges with ADV# active. During write operations, the 28F016XS-15s latch data on the rising edge of WE# if the applicable timing requirements are satisfied.

Configuration Signal

A general purpose input/output (GPIO) generates a configuration signal (CFG) input to the state machine. The configuration signal must reset to logic "0" on power-up and system reset to ensure that the operation of the state machine matches the initial configurations of the 28F016XS-15s and i960 CA microprocessor. After optimizing the 28F016XS-15s and i960 CA microprocessor, the system must drive CFG to a logic "1."

Additional 28F016XS Control Signals

The BYTE# input to the 28F016XS-15s is tied to 5.0V to configure the 28F016XS-15s for x16 mode, and A_0 and 3/5# inputs are tied to GND (A_0 is only used for byte addressing). A GPIO controls the write protect input, WP#, to the 28F016XS-15s. As shown in Figure 3, the 28F016XS-15 is compatible with either a 5.0V or a 12.0V V_{PP} voltage and is completely write protected by switching V_{PP} to GND. When V_{PP} drops below V_{PPLK}, the 28F016XS-15 will not complete program and erase operations successfully. Figure 3 also illustrates the RY/BY# output connected to a system interrupt for background erase operation. RY/BY#, WP#, and V_{PP} implementation are application dependent. See the Additional Information section of this technical paper for documentation that cover these topics in more detail.

2.2 Software Interface Considerations

Boot-Up Capability/Configuration

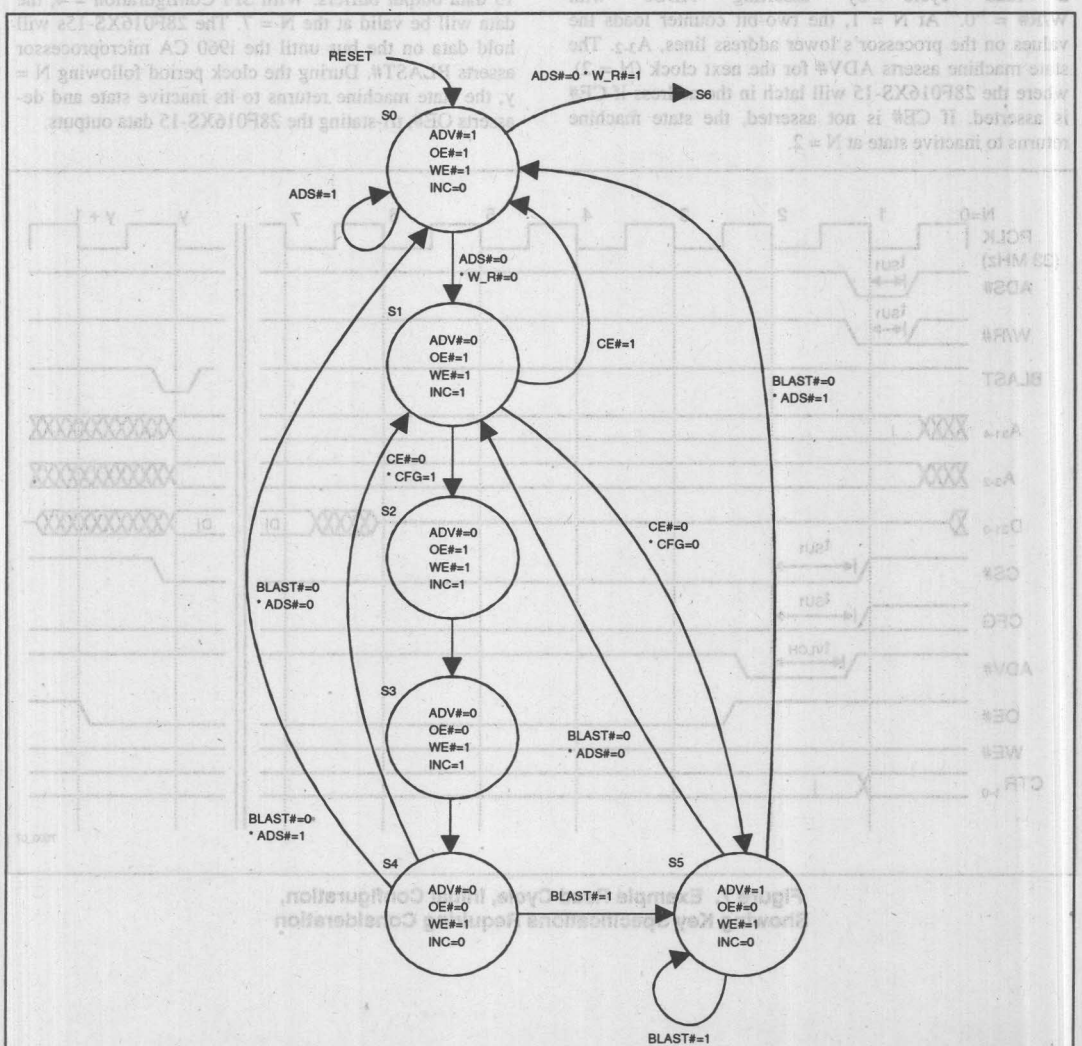
This interface supports processor boot-up from the 28F016XS memory space upon power-up or system reset. However, the boot code must follow some restrictions until it has properly configured the 28F016XS-15s, i960 CA microprocessor and CFG valve. In the default configuration state, the i960 CA microprocessor supports only non-burst reads and writes. Program control should jump to an area of RAM to execute the configuration sequence. The code will configure the 28F016XS-15s and all necessary i960 CA microprocessor programmable attributes before setting the CFG input to logic "1." Table 1 illustrates the required configuration settings for both the 28F016XS-15s and i960 CA-33 microprocessor.

Table 1. Configuration Settings for the 28F016XS-15 and i960 CA-33 Microprocessor Employing Address Pipelining at 33 MHz

Part	Parameter	Setting
28F016XS-15 (5V V_{CC})	SFI Configuration	2
i960 CA-33 Microprocessor	Ready Inputs	OFF
	Byte Ordering	LITTLE ENDIAN
	Bus Width	32-BIT
	Wait-States: Nrad	3
	Nrdd	0
	Nwad	2
	Nwdd	2
	Nxda	0
	Address Pipelining	ON
	Burst Mode	ON

2.3 Single and Burst Read Cycle Description at 33 MHz

Refer to the read cycle timing diagrams (Figures 7 and 8) and the state diagram (Figure 6) for the following read cycle discussion.



NOTE:
OE# and WE# are clocked on the inverted CLK edge

Figure 6. Read State Diagram of Single and Address Pipelined Burst Control Interface Shown in Figure 3

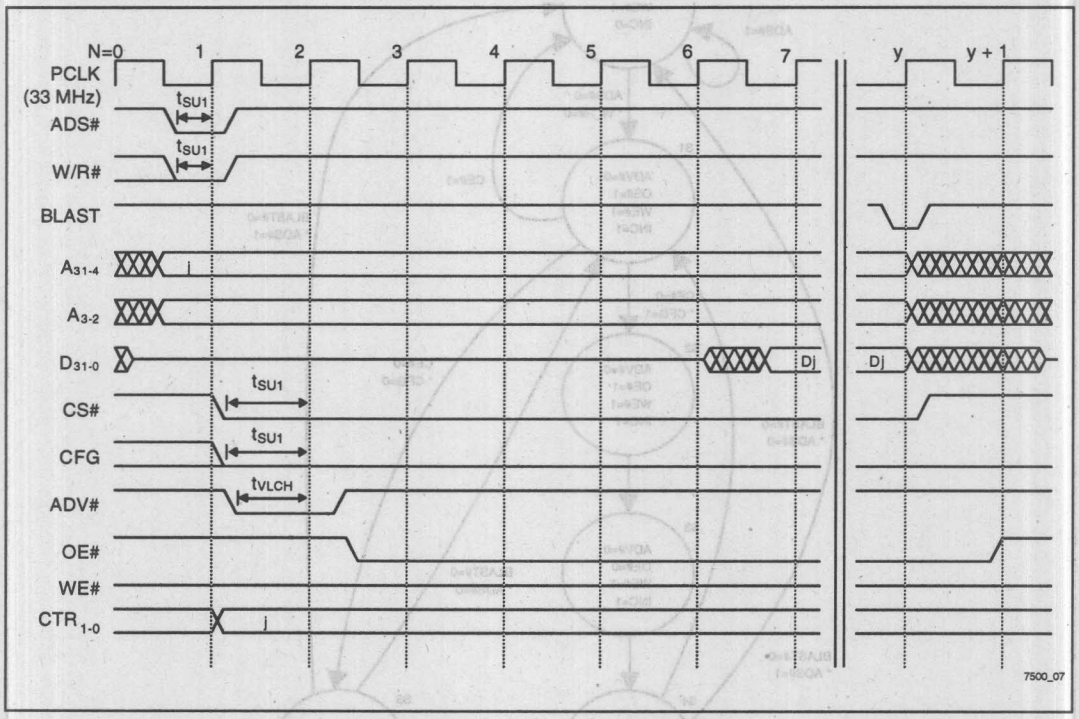


Figure 7. Example Read Cycle, Initial Configuration, Showing Key Specifications Requiring Consideration

Optimized Configuration

Figure 8 illustrates a two double-word burst read followed by a four double-word burst read with the 28F016XS-15s, i960 CA microprocessor and state machine configured for optimum read performance. With CFG = "1," the counter increments the two lower bits of the address at N = 2, N = 3 and N = 4, and ADV# remains asserted so that the 28F016XS-15 latches in four successive addresses at N = 2 through 5. With SFI Configuration = 2, the first data will be available at N = 5 for the processor to read. If a second read cycle follows the current read cycle, the i960 CA microprocessor will assert ADS# one clock after asserting BLAST#. The state machine will respond by immediately re-entering the read cycle. Otherwise, the state machine will return to its inactive state waiting for a new access targeting the 28F016XS memory space.

When implementing the i960 CA microprocessor address pipelining capability, the state machine controlling CE# monitors the upper address lines, ADS# and BLAST#. CE# is held active upon detecting an access targeting the flash memory space until BLAST# is asserted with ADS# de-asserted. When BLAST# and ADS# are active at the same time, a pipelined read access is in progress. If the current pipelined access targets the 28F016XS memory space, the CE# state machine will hold CE# active until BLAST# is sampled active again.

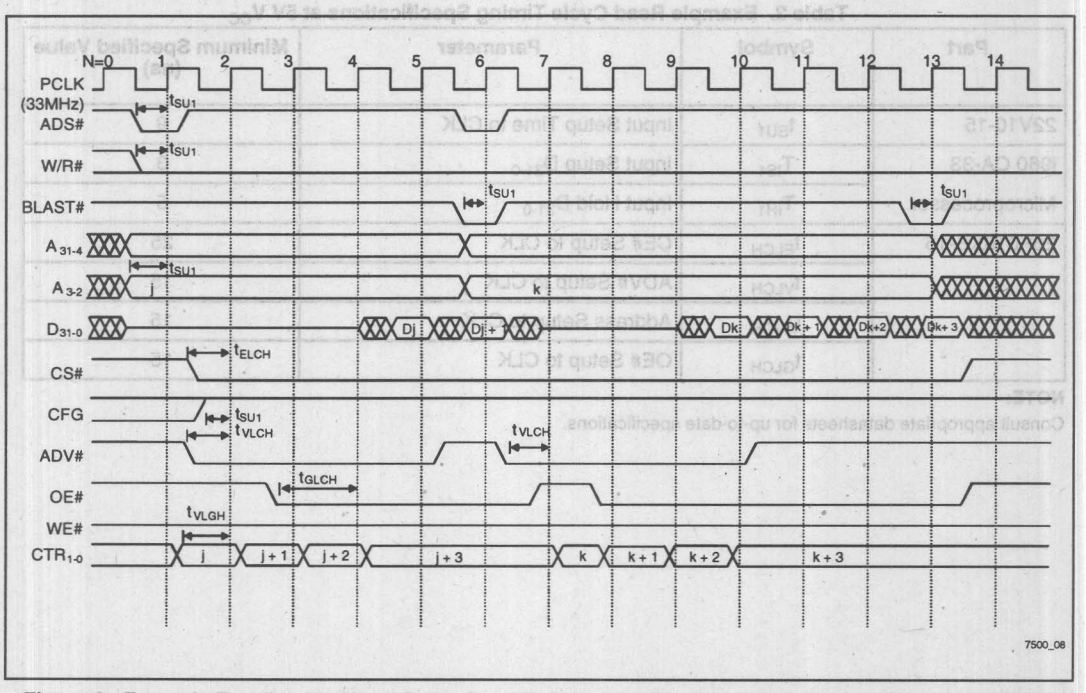


Figure 8. Example Two Double-Word Burst Read Followed by Pipelined Four Double Word Burst Read Showing Key Specifications Requiring Consideration

Critical Timings

Table 2 describes the critical timings illustrated in Figures 7 and 8. One such critical timing is the data hold time. The i960 CA-33 microprocessor requires a 5 ns hold time after the clock edge. The 28F016XS-15 guarantees a 5 ns data hold after clock, meeting the processor's hold requirement with 0 ns of margin.

This design provides 7 ns of margin to meet the 3 ns setup time of the i960 CA-33 microprocessor data inputs, outputting data t_{CHQV} after a rising CLK edge.

Another critical timing concerns CE# during pipelined read accesses. Since the 28F016XS-15 specifies zero data hold from CE# going high, the chip select state machine must hold CE# active for 5 ns to satisfy the i960 CA-33 microprocessor data input hold specification of 5 ns. Hence, the chip select state machine holds CE# active for an additional clock period after detecting BLAST# active.

The i960 CA-33 microprocessor control outputs ADS# and W/R# have 3 ns of margin and BLAST# has 5 ns of margin to meet the 22V10-15 input setup requirement.

Consult the appropriate datasheets for complete timing information.

Table 2. Example Read Cycle Timing Specifications at 5V V_{CC}

Part	Symbol	Parameter	Minimum Specified Value (ns)
22V10-15	t_{SU1}	Input Setup Time to CLK	9
i960 CA-33	T_{IS1}	Input Setup D_{31-0}	3
Microprocessor	T_{IH1}	Input Hold D_{31-0}	5
28F016XS-15	t_{ELCH}	CE# Setup to CLK	25
	t_{VLCH}	ADV# Setup to CLK	15
	t_{AVCH}	Address Setup to CLK	15
	t_{GLCH}	OE# Setup to CLK	15

NOTE:
Consult appropriate datasheets for up-to-date specifications.

2.4 Single and Burst Write Cycle Description at 33 MHz

Refer to the write cycle timing diagrams and the state diagram (Figure 9) for the following write cycle discussion.

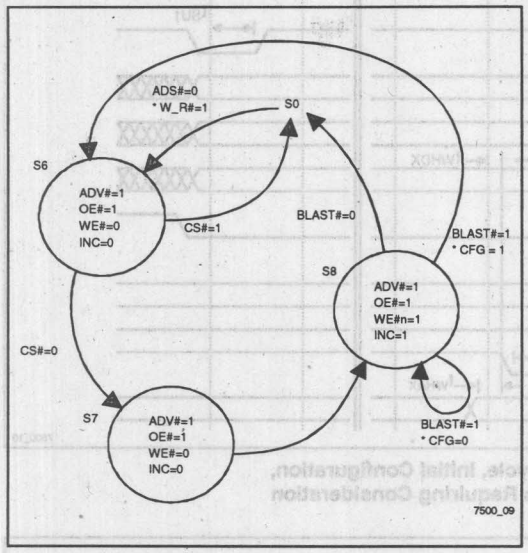


Figure 9. Write State Diagram for the Interface Logic Shown in Figure 3

Initial Configuration

Figure 10 illustrates a write cycle. In the reset/power-up configuration state, the interface supports only non-burst writes. The i960 CA microprocessor initiates a write cycle by asserting ADS# with W/R# = "1." At N = 1, the two-bit counter loads the values on address bits A_{3-2}. The state machine asserts WE# (to meet timing requirements, WE# is falling-edge triggered) on the falling edge between N = 1 and N = 2. WE# remains asserted for two clock periods, in order to meet the 28F016XS-15 timing requirements. The state machine then enters a holding state until the processor asserts BLAST#, after which time the interface state machine will return to an idle state.

Optimized Configuration

Figure 11 illustrates an optimized two double-word burst write cycle. When the first data write is completed at N = 4, the counter increments the two lower address bits, and the state machine asserts WE# on the next falling clock edge to begin the next the 28F016XS-15 data write. The i960 CA microprocessor must provide the next data during the clock period following N = 4. The data writes continue to the next consecutive addresses until the i960 CA microprocessor asserts BLAST#, indicating the end of the burst write cycle.

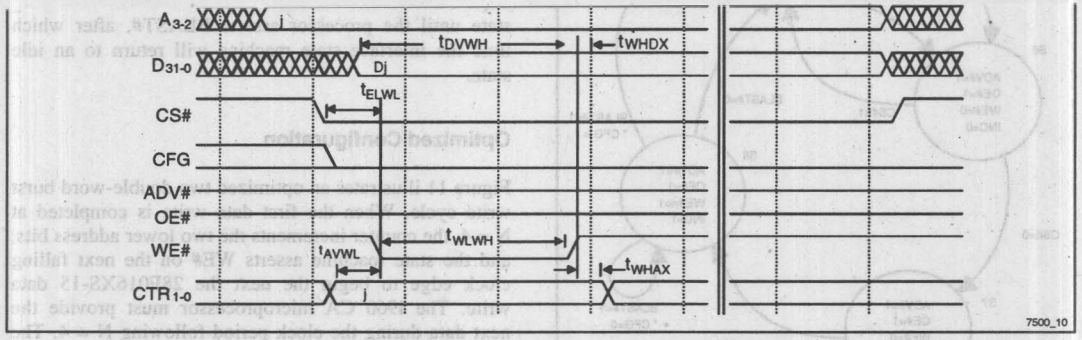


Figure 10. Example Write Cycle, Initial Configuration, Showing Key Specifications Requiring Consideration

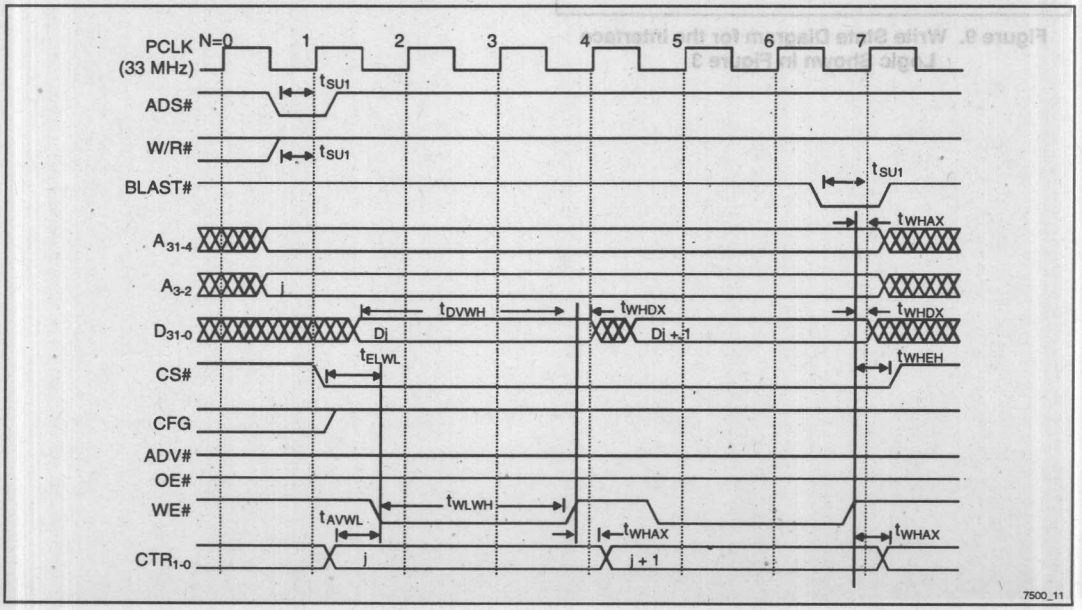


Figure 11. Example Two Double-Word Burst Write Illustrating Key Specifications Requiring Considerations

Critical Timings

Table 3 describes the critical timings illustrated in Figures 10 and 11.

One critical hold time to notice is t_{WHAX}. WE# is guaranteed to transition within 8 ns from the falling clock edge. Therefore, the t_{WHAX} requirement has 2 ns of margin for CTR_{1-0} and 5 ns of margin for A_{31-4} inputs.

Also notice that CTR_{1-0} must be valid before WE# is asserted. CTR_{1-0} are guaranteed valid 8 ns after the rising clock edge, providing 9 ns of margin.

Consult the appropriate datasheets for full timing information.

Table 3. Example Write Cycle Timing Parameters at 5V V_{CC}

Part	Symbol	Parameter	Minimum Specified Value (ns)
22V10-15	t_{SU1}	Input Setup Time to CLK	9
28F016XS-15	t_{ELWL}	CE# Setup to WE# Going Low	0
	t_{AVWL}	Address Setup to WE# Going Low	0
	t_{WLWH}	WE# Pulse Width	50
	t_{DVWH}	Data Setup to WE# Going High	50
	t_{WHDX}	Data Hold from WE# High	0
	t_{WHAX}	Address Hold from WE# High	5
	t_{WHEH}	CE# Hold from WE# High	5

NOTE:
Consult appropriate datasheets for up-to-date specifications.

3.0 Optimized i960 JF MICROPROCESSOR INTERFACE

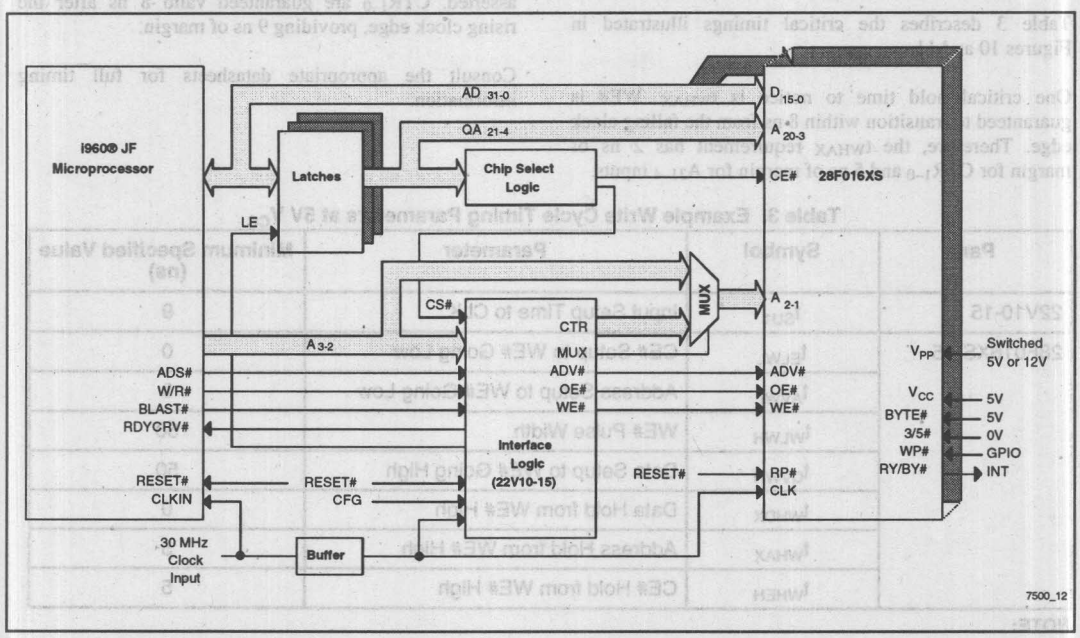


Figure 12. Minimal Interface Logic Required in Interfacing the 28F016XS-15 to the i960 JF-33 Microprocessor to Sustain 2-0-0-0 Burst Read Performance up to 30 MHz

Using this particular design, the 28F016XS interface to the i960 JF-33 microprocessor can achieve 2-0-0-0 wait-state read performance up to 30 MHz, supporting burst transfers. Contact your Intel or distribution sales office for schematic and PLD files for the design documented in the section.

See Section 4.0 for an alternative i960 JF design.

3.1 Circuit Description

This design, illustrated in Figure 12, uses two 28F016XS-15s to match the 32-bit data bus of the i960 JF-33 microprocessor. Four octal latches, enabled by the LE signal, de-multiplex the 32-bit address from the AD bus. The latched address bits QA_{21-4} and the counter outputs CTR_{1-0} from the PLD select locations within the 28F016XS memory space. A two-bit counter implemented in the PLD loads and increments the processor's lower address lines at the beginning of each memory cycle. The processor supplies the 28F016XS-15s with the initial address during a read transaction, and the counter provides the subsequent burst addresses for the remaining duration of the read transaction.

CLK Option

A 30 MHz clock signal drives the i960 JF microprocessor CLKIN input. The buffer delays the system CLK and drives the PLD and 28F016XS-15s. The buffer introduces an intentional system clock skew, which provides additional time for the processor to meet the 28F016XSs' address setup time on the initial memory access. At a slower operating frequency, this intentional delay may not be required to satisfy the 28F016XSs' address setup specification.

Multiplexer (MUX)

To achieve this wait-state profile, the processor loads the 28F016XSs with the initial address directly. The interface logic enables the MUX to permit the processor's lower address lines, A_{3-2}, access to the lower flash memory address lines during the initial access of a burst or single read transaction. Next, the interface logic switches the data flow path through the MUX. The integrated two-bit

counter then takes over, driving the 28F016XSs' A_{2-1} address inputs. The counter supplies the flash memory with consecutive burst addresses for the remaining duration of the read transaction.

Reset

An active-low reset signal, RESET#, drives the RESET# inputs of the i960 JF microprocessor and PLD and 28F016XS-15s. Figure 5 illustrates a suggested logic configuration for generating RESET#.

Interface Control Signals

The processor's ADS# and W/R# signals, just as in the i960 CA microprocessor design, serve as inputs to the state machine. The state machine controls an integrated two-bit counter and generates the OE#, WE# and ADV# signals for the 28F016XS-15s. The state machine also generates the RDYRCV# signal to the i960 JF microprocessor to control the wait-state profile.

Configuration Signal

A general purpose input/output (GPIO) generates the configuration signal (CFG) for input to the state machine. The configuration signal must be reset to logic "0" on power-up and system reset to ensure coherency between the state machine and 28F016XS-15s. After optimizing the 28F016XS-15s, the reconfiguration signal must switch to a logic "1" to take advantage of the new configuration.

Additional Control Signals

For information regarding 3/5#, BYTE#, WP#, RY/BY# and Vpp, see Section 2.1.

3.2 Software Interface Considerations

Boot-Up Capability

This interface supports processor boot-up from the 28F016XS-15 memory space after power-up or system reset. The initial system configuration will execute 4-1-1-1 wait-state read performance until the SFI Configuration is set to 2 and CFG input is set to logic "1." Program control should jump to an area of RAM to execute the configuration sequence. A pseudocode flow for this configuration sequence is shown below.

Execute Device Configuration command sequence
Activate CFG signal
End

The SFI Configuration must be set to 2 before the CFG input is set to logic "1." Thereafter, burst read wait-state performance will improve to 2-0-0-0.

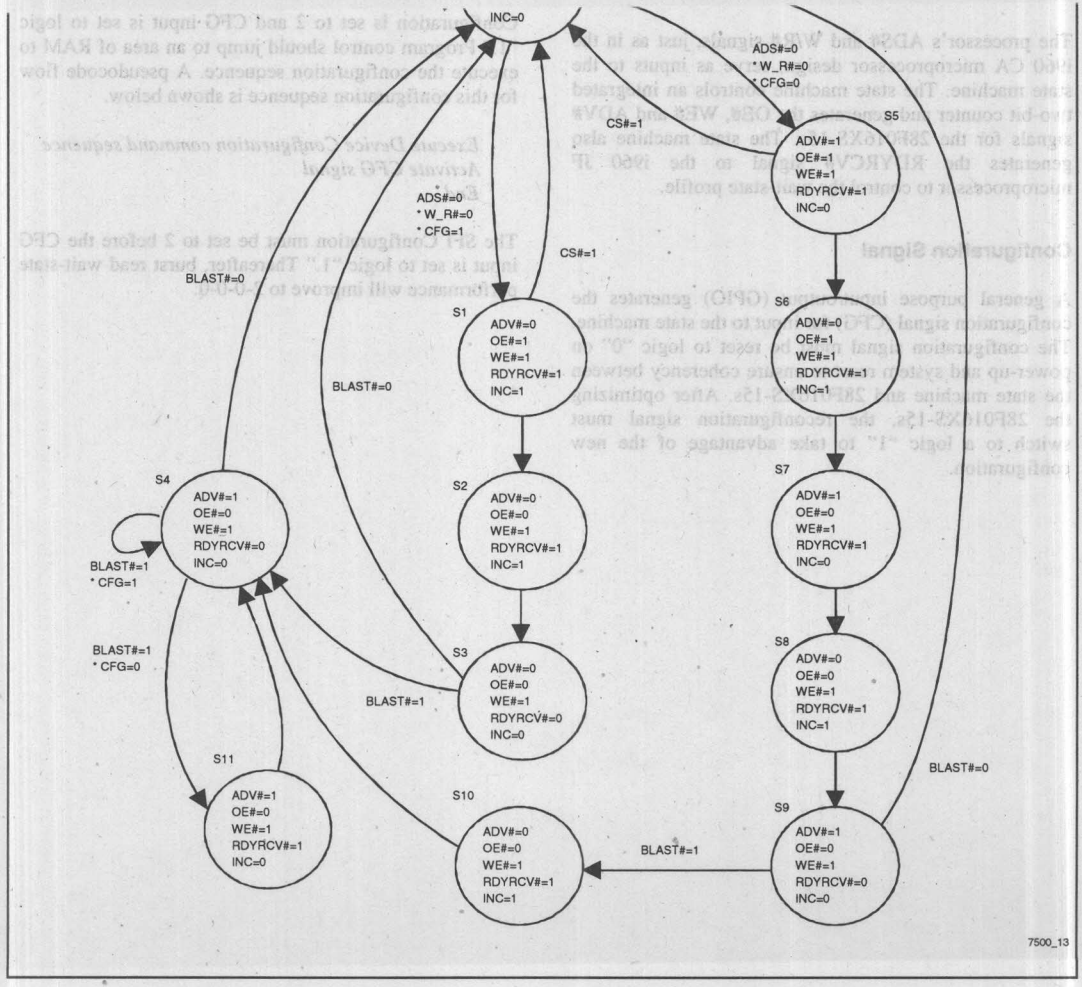


Figure 13. Read State Diagram of Burst Control Interface Shown in Figure 12

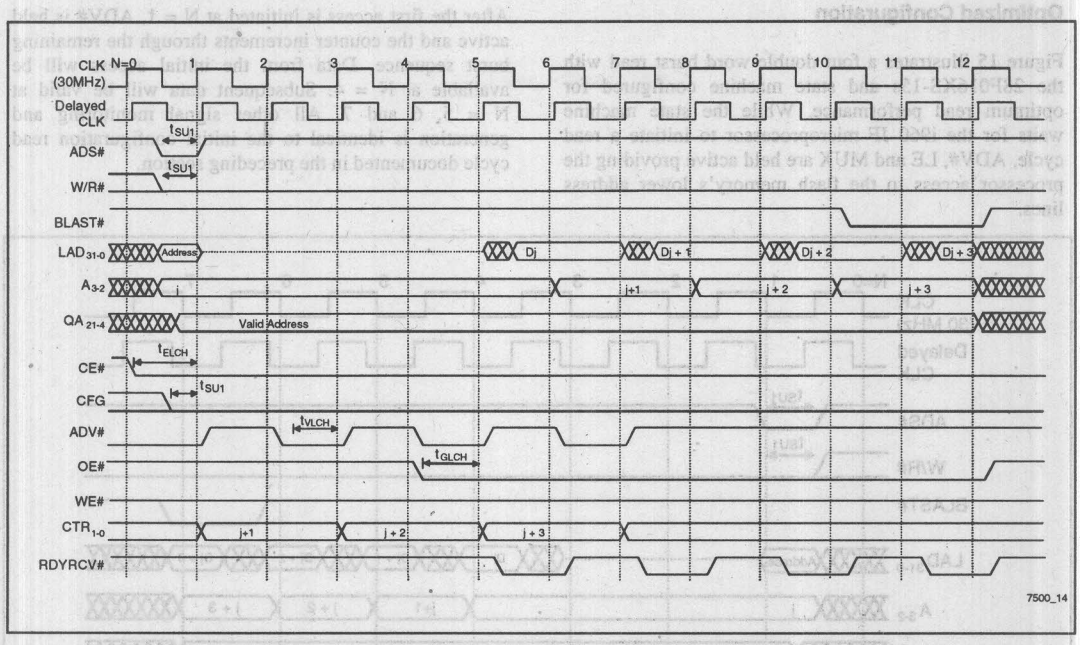


Figure 14. Example Four Double-Word Burst Read in Initial Configuration Showing Key Specifications Requiring Consideration

Initial Configuration

Figure 14 illustrates a four double-word burst read cycle with the 28F016XS-15s and state machine in a reset/power-up configuration state.

Initially, the interface logic drives LE, ADV# and MUX active while waiting for the processor to initiate a read cycle targeting the 28F016XSs. With the MUX active, the processor drives the lower flash memory address lines in anticipation of a flash memory access. During this anticipation state, CE# is active to prevent a t_{ELCH} violation on the first access initiated by the processor. The delayed CLK also prevents a possible timing violation from occurring by providing the processor with sufficient time to meet the 28F016XSs' t_{AVCH} specification.

When the i960 JF microprocessor initiates a read cycle by asserting ADS# with W/R# = "0," the state machine loads and increments the two-bit counter. The counter is incremented upon the initial load sequence because the processor supplies the flash memory with the initial address.

The state machine will de-assert ADV# at N = 1 and switches the data flow path through the MUX. The two-bit counter then drives the flash memory's lower address lines for the remaining duration of the read transaction. The state machine asserts ADV# at N = 3 to load the next read address into the 28F016XS-15s. De-asserting ADV# for one clock cycle (at N = 2, 4, 6 and 8) between accesses forces the 28F016XS-15s to hold data output for two clock cycles (access stretching), which allows time for the data to stabilize and meet the timing requirements of the i960 JF microprocessor bus.

While ADV# is asserted, the counter increments, providing the flash memory with successive burst addresses. The state machine asserts OE# at N = 4 to enable the 28F016XS-15 data output buffers. With the SFI Configuration = 4, the data will be valid at the i960 JF microprocessor data inputs at N = 6, 8, 10 and 12.

The state machine asserts RDYRCV# to inform the i960 JF microprocessor that data is valid. RDYRCV# is returned active at N = 6, 8, 10 and 12. Upon detecting BLAST# active, the interface will transition to its idle state where it waits for the start of a new bus cycle.

Optimized Configuration

Figure 15 illustrates a four double-word burst read with the 28F016XS-15s and state machine configured for optimum read performance. While the state machine waits for the i960 JF microprocessor to initiate a read cycle, ADV#, LE and MUX are held active providing the processor access to the flash memory's lower address lines.

After the first access is initiated at N = 1, ADV# is held active and the counter increments through the remaining burst sequence. Data from the initial access will be available at N = 4. Subsequent data will be valid at N = 5, 6 and 7. All other signal monitoring and generation is identical to the initial configuration read cycle documented in the preceding section.

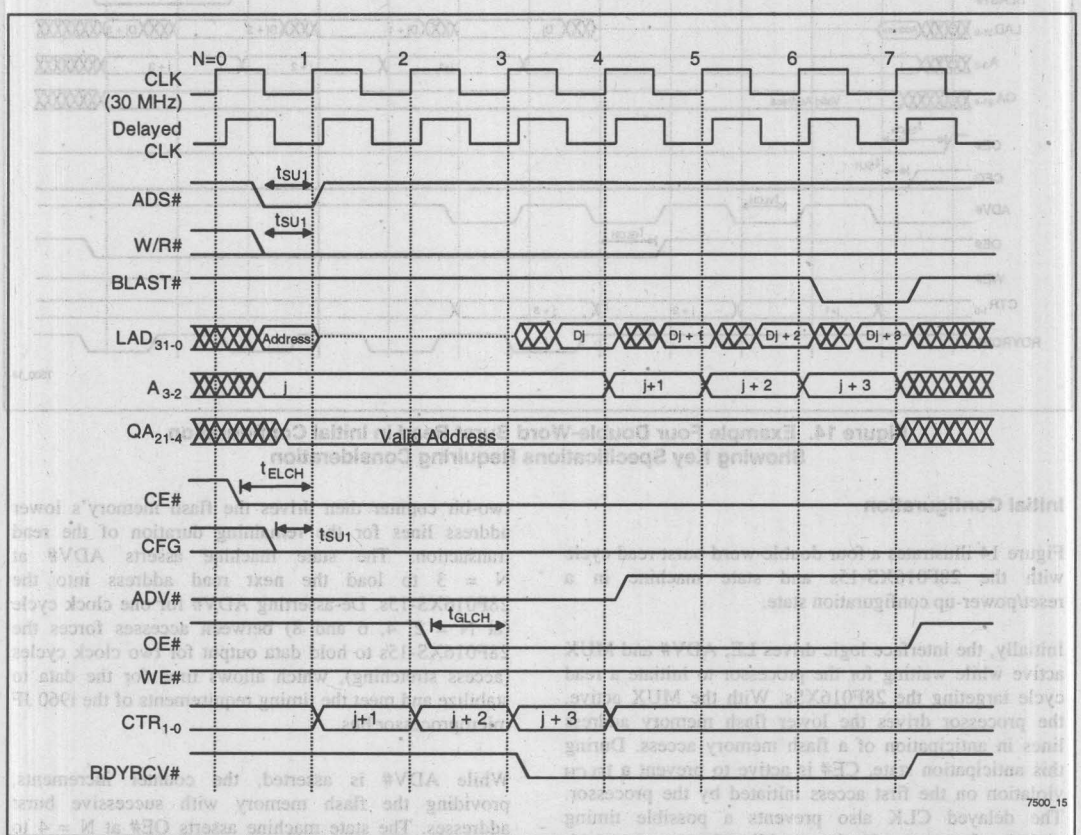


Figure 15. Example Four Double-Word Burst Read Illustrating Important Timing Parameters Requiring Consideration

Critical Timings

In these two configurations, there are some critical timings that need to be examined. Table 4 depicts these critical timings which are illustrated in Figures 14 and 15.

First, the buffer delay can cause timing violations if not chosen correctly. The purpose of the buffer is to provide sufficient time for the processor to load the 28F016XSs with the initial address during a read transaction. Therefore, the buffer must have a minimum delay that satisfies the flash memory's t_{AVCH}.

$$\text{Latch Delay} - t_{OV1} - t_{AVCH} - 1/\text{CLKIN} = \text{Buffer Delay}$$

As the previous equation illustrates, the minimum buffer delay is dependent upon several different variables: CLKIN frequency and latch delay. At a slow operation frequency, the interface does not require a buffer delay.

The buffer can also affect the processor's data setup time. Hence, the buffer must have a maximum delay of no greater than:

$$1/\text{CLKIN} - t_{CHQV} - t_{IS1} = \text{Buffer Delay}$$

Consult the appropriate datasheets for complete timing information.

Table 4. Example Read Cycle Timing Parameters at 5V V_{CC}

Part	Symbol	Parameter	Minimum Specified Value (ns)
Buffer	$t_{PHL,PLH}$	Buffer Delay	1.5
22V10-15	t_{SU1}	Input Setup Time to CLK	9
28F016XS-15	t_{ELCH}	CE# Setup to CLK	25
	t_{VLCH}	ADV# Setup to CLK	15
	t_{AVCH}	Address Setup to CLK	15
	t_{GLCH}	OE# Setup to CLK	15

NOTE:
Consult appropriate datasheets for up-to-date specifications.

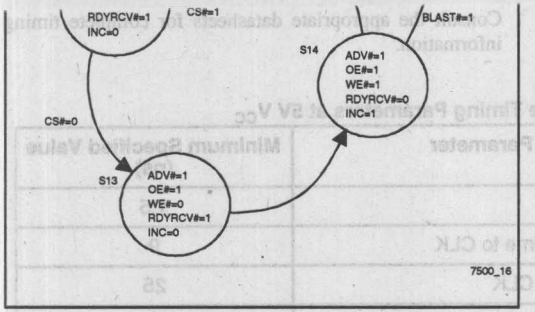


Figure 16. Write State Diagram of Burst Control Interface Shown in Figure 12

loads the values on the processor's lower address lines. The state machine asserts WE# (to meet timing requirements, WE# is falling-edge triggered) on the falling edge between N = 1 and N = 2. WE# remains asserted for four clock periods, in order to meet 28F016XS-15 timing requirements. The state machine asserts RDYRCV# for N = 4 to inform the i960 JF microprocessor to supply the next data. At N = 4, the counter increments the two lower address bits, and the state machine asserts WE# on the next falling clock edge to begin the next data write to the 28F016XS-15s. The data writes continue until the processor asserts BLAST#, noting the end of the current write transaction. The SFI Configuration has no effect on the write cycle.

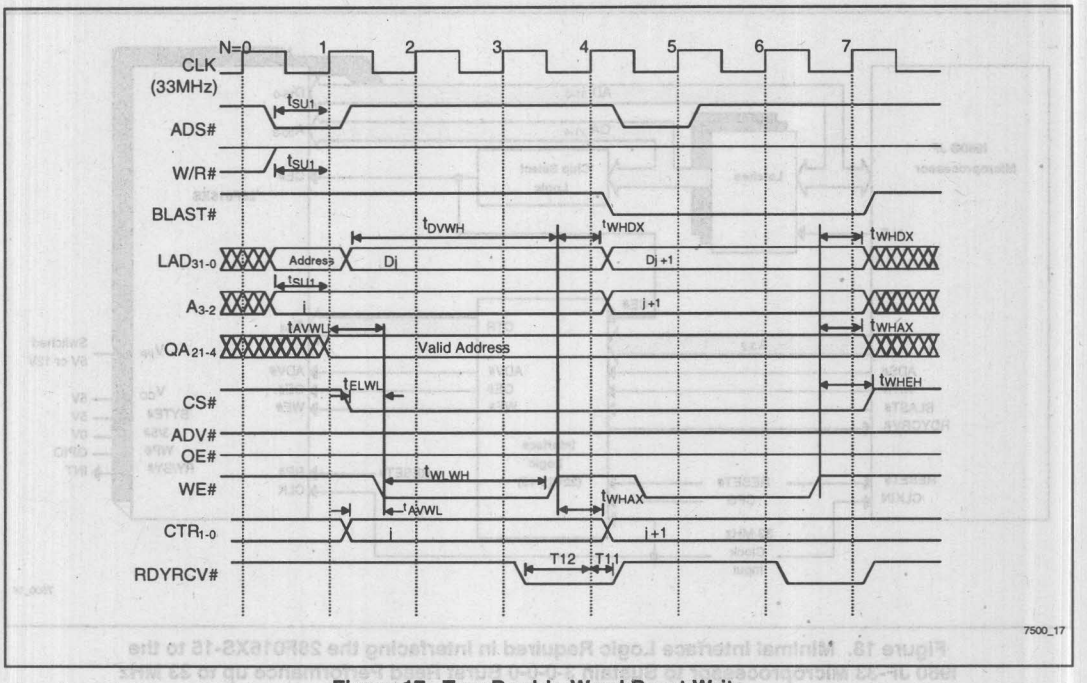


Figure 17. Two Double-Word Burst Write

Critical Timings

Table 5 describes the critical timings illustrated in Figure 17.

Notice that CTR_{1-0}, and CS# must be valid before WE# is asserted. CTR_{1-0} are guaranteed to be valid 8 ns after the rising clock edge, providing 12 ns of margin.

Consult the appropriate datasheets for full timing information.

Table 5. Example Write Cycle Timing Parameters at 5V V_{CC}

Part	Symbol	Parameter	Minimum Specified Value (ns)
22V10-15	t_{SU1}	Input Setup Time to CLK	9
28F016XS-15	t_{ELWL}	CE# Setup to WE# Going Low	0
	t_{AVWL}	Address Setup to WE# Going Low	0
	t_{WLWH}	WE# Pulse Width	50
	t_{DVWH}	Data Setup to WE# Going High	50
	t_{WHDX}	Data Hold from WE# High	0
	t_{WHAX}	Address Hold from WE# High	5
	t_{WHEH}	CE# Hold from WE# High	5

NOTES:
Consult appropriate datasheets for up-to-date specifications.

4.0 Standard i960 JF MICROPROCESSOR INTERFACE

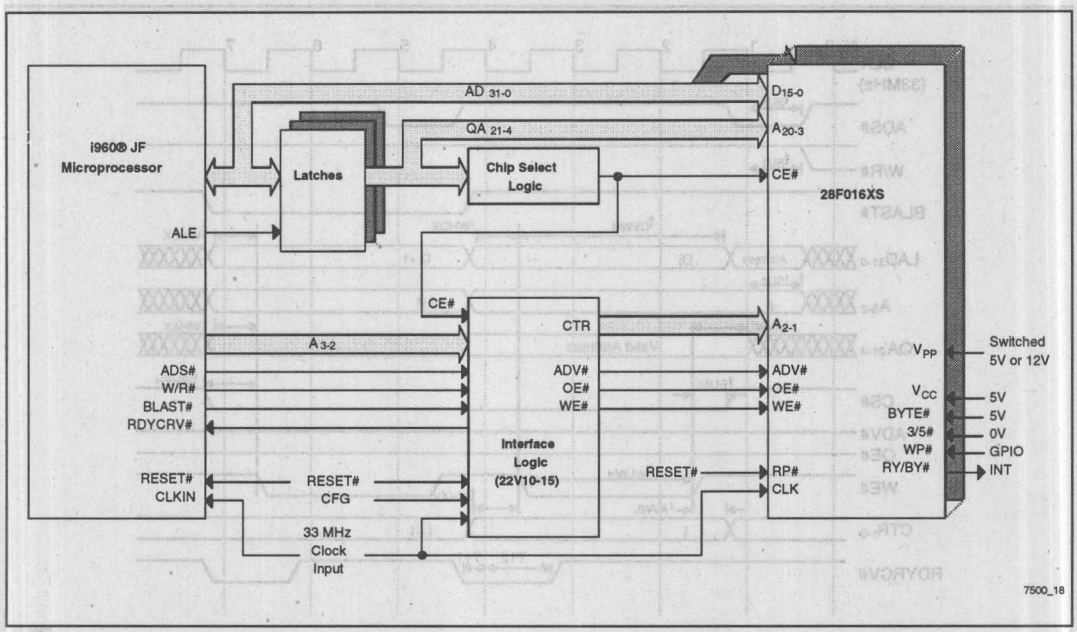


Figure 18. Minimal Interface Logic Required in Interfacing the 28F016XS-15 to the i960 JF-33 Microprocessor to Sustain 3-0-0-0 Burst Read Performance up to 33 MHz

The 28F016XS-15 interface to the i960 JF-33 microprocessor, illustrate in Figure 18, delivers 3-0-0-0 wait-state read performance up to 33 MHz. The design requires only one 22V10 to handle all interfacing requirements. Contact your Intel or distribution sales office for schematic and PLD files for the interface documented in this section.

See Section 3.0 for an alternative design.

4.1 Circuit Description

This interface is very similar to the optimized i960 JF design described in Section 3.0. This design however, eliminates the buffer and multiplexer requirement. For specific circuitry information involved in this standard interface, reference Section 3.0.

CLK Option

Unlike the optimized design in Section 3.0, a buffer is not implemented in this interface. The 33 MHz system clock drives the processor, PLD and flash memory. To reduce system clock skew, position the PLD and 28F016XSs within close proximity to the microprocessor.

4.2 Software Interface Considerations

This interface supports processor boot-up from the flash memory space after power-up or system reset. Initially, the read wait-state performance will be 5-1-1-1 until the SFI Configuration value is modified. The 28F016XS operates at optimal performance with a SFI Configuration value of 2 at 33 MHz. Program control should jump to an area of RAM to execute a configuration sequence which optimizes the flash memory and interface state machine for the given operating frequency.

4.3 Burst Read Cycle Description at 33 MHz

Refer to the read cycle timing diagrams and state diagram (Figure 19) for the following discussions of the read cycle.

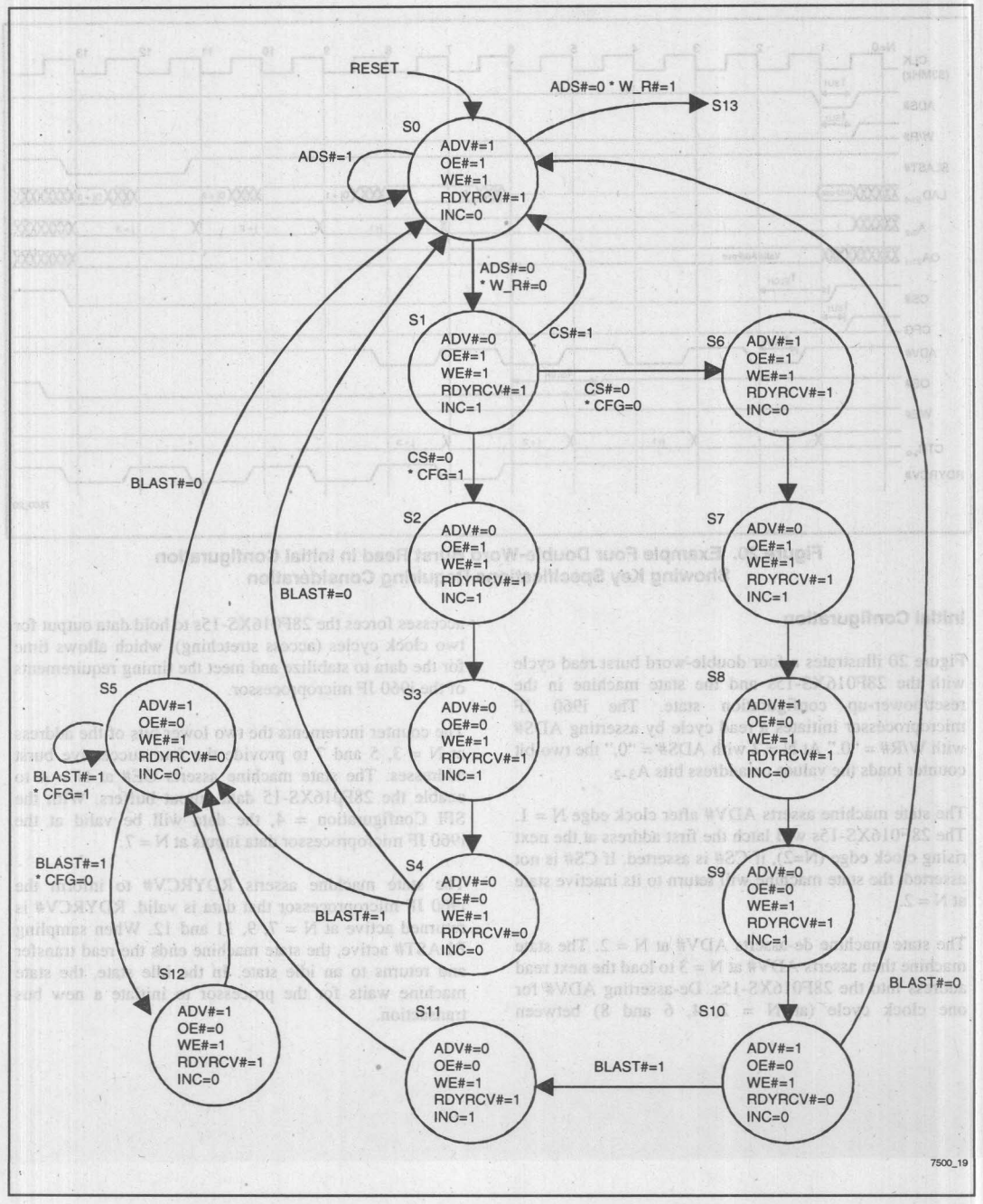


Figure 19. Read State Diagram of Burst Control Interface Shown in Figure 18

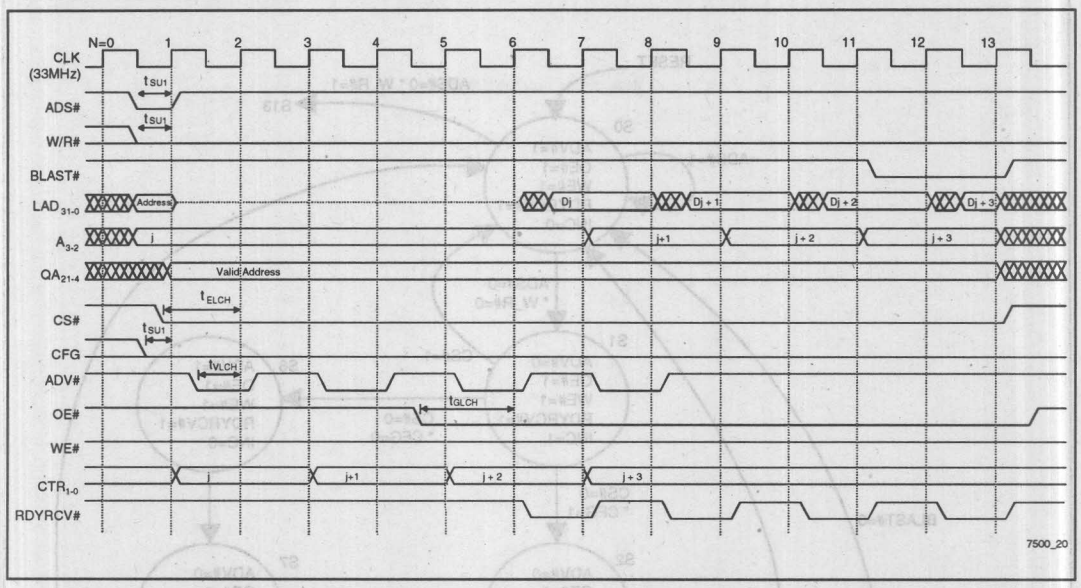


Figure 20. Example Four Double-Word Burst Read in Initial Configuration Showing Key Specifications Requiring Consideration

Initial Configuration

Figure 20 illustrates a four double-word burst read cycle with the 28F016XS-15s and the state machine in the reset/power-up configuration state. The i960 JF microprocessor initiates a read cycle by asserting ADS# with W/R# = "0." At N = 2 with ADS# = "0," the two-bit counter loads the values on address bits A_{3-2}.

The state machine asserts ADV# after clock edge N = 1. The 28F016XS-15s will latch the first address at the next rising clock edge (N=2), if CS# is asserted. If CS# is not asserted, the state machine will return to its inactive state at N = 2.

The state machine de-asserts ADV# at N = 2. The state machine then asserts ADV# at N = 3 to load the next read address into the 28F016XS-15s. De-asserting ADV# for one clock cycle (at N = 2, 4, 6 and 8) between accesses forces the 28F016XS-15s to hold data output for two clock cycles (access stretching), which allows time for the data to stabilize and meet the timing requirements of the i960 JF microprocessor.

The counter increments the two lower bits of the address at N = 3, 5 and 7 to provide the four successive burst addresses. The state machine asserts OE# at N = 4 to enable the 28F016XS-15 data output buffers. With the SFI Configuration = 4, the data will be valid at the i960 JF microprocessor data inputs at N = 7.

The state machine asserts RDYRCV# to inform the i960 JF microprocessor that data is valid. RDYRCV# is returned active at N = 7, 9, 11 and 12. When sampling BLAST# active, the state machine ends the read transfer and returns to an idle state. In the idle state, the state machine waits for the processor to initiate a new bus transaction.

Optimized Configuration

Figure 21 illustrates a four double-word burst read with the 28F016XS-15s and state machine configured for optimum read performance. With the SFI Configuration = 2, ADV# is held active and the counter increments at N = 2, 3 and 4, supplying the 28F016XS-15s with four consecutive accesses. Data from the initial access will be valid for transfer at N = 5. Subsequent data will be valid at N = 6, 7 and 8, improving the read wait-state performance to 3-0-0-0. All other signal monitoring and generation are identical to the reset/power-up configuration read cycle documented in the preceding section.

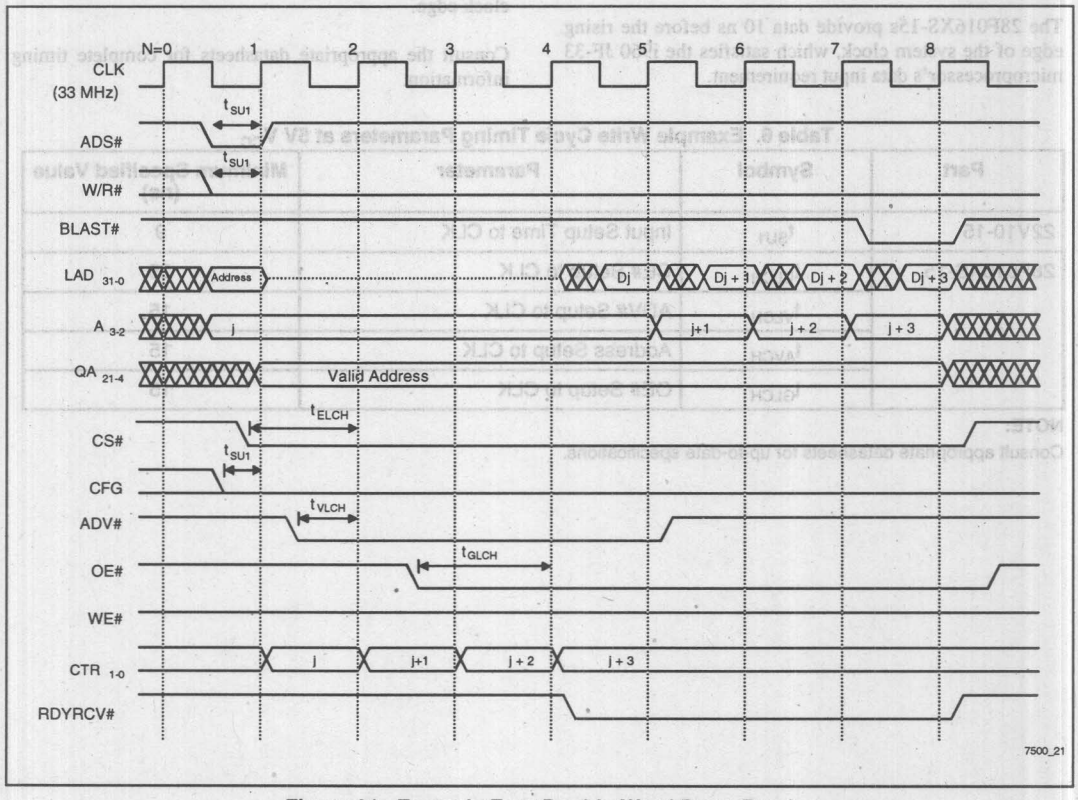


Figure 21. Example Four Double-Word Burst Read Illustrating Important Timing Parameters Requiring Consideration

Part	Symbol	Parameter	Minimum Specified Value (ns)
22V10-15	t_{SU1}	Input Setup Time to CLK	9
28F016XS-15	t_{ELCH}	CE# Setup to CLK	25
	t_{VLCH}	ADV# Setup to CLK	15
	t_{AVCH}	Address Setup to CLK	15
	t_{GLCH}	OE# Setup to CLK	15

NOTE:
Consult appropriate datasheets for up-to-date specifications.

4.4 Burst Write Cycle Description at 33 MHz

The write interface for this design behaves similar to the optimized design's write interface described in Section 3.4. The only difference between the two designs is the absence of the MUX. Therefore, this design's write state machine does not have to concern itself with controlling the MUX data flow. Instead, the interface simply loads and holds the address presented by the processor for the duration of the write operation.

For further details information about this cycle and write timing waveform, refer to Section 3.4.

5.0 INTERFACING TO OTHER i960 MICROPROCESSORS

i960 CF/HA/HD/HT-16, i960 CF/HA/HD/HT-25 and i960 CF/HA/HD-33 Microprocessors

The i960 CF/HA/HD/HT microprocessor bus interfaces are compatible with the i960 CA microprocessor bus interface. Therefore, the 28F016XS-15 interfaces described in Section 2.0 for the i960 CA-33 microprocessor work equally well with these i960 micoprocessors at 25 and 33 MHz.

At 16 MHz, the interface requires a slight modification because the SFI Configuration value at 16 MHz equals 1. Therefore, the 28F016XS-15 will drive data one CLK period after initiating a read access. The interface will return READY# to the microprocessor one CLK cycle earlier as opposed to the i960 CA-33 design. Therefore, the 28F016XS-15 interface to these microprocessors, operating at 16 MHz, will deliver 2-0-0-0-1-0-0-0. . .wait-state read performance.

i960 KA and i960 KB Microprocessors

The 28F016XS interface to the i960 Kx microprocessor series will be very similar to the i960 JF interface described in Sections 3.0 and 4.0. The only difference between the two designs is the i960 Kx's lack of a BLAST# output signal. The i960 JF interfaces use this signal to determine the end to the burst transaction. Since the i960 Kx microprocessor does not have a BLAST# signal, the interface logic must examine the processor's lower address lines to determine the length of the read transaction.

For further detailed information about this interface, please reference the i960 JF interfaces. Contact your Intel or distribution sales office for schematic and PLD files for the 28F016XS interface to the i960 Kx microprocessor.

i960 SA Microprocessor, i960 SB Microprocessor

The 28F016XS's interface to the i960 Sx microprocessor series will be similar to the i960 JF microprocessor interfaces, with the following differences:

- The i960 Sx microprocessor series has a 16-bit data bus multiplexed with the lower 16 of 32 address bits. Therefore, a single 28F016XS will match the width of the data bus.
- The i960 Sx microprocessor series supports eight double-word burst transfers. Therefore, the 28F016XS interface will require a three-bit counter to generate the lower three bits of the burst addresses.

6.0 CONCLUSION

This technical paper has described the interface between the 28F016XS 16-Mbit flash memory component and the i960 microprocessor family. These designs require minimal glue logic while achieving exceptional read performance. For further information about the 28F016XS, reference the Additional Information section of this technical paper. Please contact your local Intel or distribution sales office for more information on Intel's flash memory products.

ADDITIONAL INFORMATION

Order Number	Document/Tools
297372	16-Mbit Flash Product Family User's Manual
290532	28F016XS Datasheet
297500	"Interfacing 28F016XS to the Intel486™ Microprocessor Family"
292147	AP-398, "Designing with the 28F016XS"
292146	AP-600, "Performance Benefits and Power/Energy Savings of 28F016XS Based System Designs"
292163	AP-610, "Flash Memory In-System Code and Data Update Techniques"
292165	AB-62, "Compiled Code Optimizations for Flash Memories"
297508	FLASHBuilder Utility
Contact Intel/Distribution Sales Office	28F016XS Benchmark Utility
Contact Intel/Distribution Sales Office	28F016XS iBIS Model
Contact Intel/Distribution Sales Office	28F016XS VHDL Model
Contact Intel/Distribution Sales Office	28F016XS Timing Designer Library File
Contact Intel/Distribution Sales Office	28F016XS Orcad and ViewLogic Schematic Symbols

REVISION HISTORY

Number	Description
001	Original Version
002	Added Optimized i960 JF Microprocessor Interface That Delivers 2-0-0-0 Wait-State Performance up to 30 MHz Removed Detailed i960 KB Microprocessor Interface Description Added i960 KB Microprocessor Interfacing Guidelines to Section 5.0, "Interfacing to Other i960 Microprocessors"
003	Added 3/5# Pin to Block Diagrams Modification made to Figure 5 in regards to recommended reset component (MAX705 changed to MAX706) Change to Figure 21, showing an optimized read at 33 MHz. Cosmetic Changes

APPENDIX A
PLD FILES

PLD file for the standard 28F016XS interface to the i960 CA Microprocessor described in Section 2.0.

```
Title           28F016XS / i960® CA Microprocessor Interface State Machine
Pattern         PDS
Revision        1
Authors         Example
Company         Intel Corporation - Folsom, California
Date            1-25-94

CHIP      STATEMACHINE      85C22V10

; inputs
PIN   1           CLK
PIN               ADS_n             ; address status - i960 CA microprocessor
PIN               W_R_n             ; W/R# - i960 CA microprocessor
PIN               BLAST_n           ; burst last - i960 CA microprocessor
PIN               CS_n              ; chip select - 28F016XS
PIN               CFG               ; 28F016XS/i960 CA microprocessor config status set input
PIN               A2                ; LAD bit 2
PIN               A3                ; LAD bit 3
PIN               RESET             ; resets all FFs in device
PIN   25          GLOBAL            ; virtual pin to implement reset

; outputs
PIN               CTR0              ; burst counter out - 28F016XS-A1
PIN               CTR1              ; burst counter out - 28F016XS-A2
PIN               /WE               ; write enable - 28F016XS
PIN               /OE               ; output enable - 28F016XS
PIN               Q0                ; state variables
PIN               Q1
PIN               /ADV              ; state variable and address valid - 28F016XS
PIN               Q3

 ; burst counter control signals
STRING  LD '(/ADS_n)'                       ; load
STRING  INC '(/ADV + ADV * Q3 * Q1 * Q0)'   ; increment

STATE  MOORE_MACHINE
DEFAULT_BRANCH S0

; state assignments
          S0 =    /Q3 * /ADV * /Q1 * /Q0
          S1 =    /Q3 *  ADV * /Q1 * /Q0
          S2 =     Q3 *  ADV * /Q1 * /Q0
          S3 =     Q3 *  ADV * /Q1 *  Q0
          S4 =    /Q3 *  ADV * /Q1 *  Q0
```

```
S1 :=   (/CS_n * /CFG)          -> S5     ; 28F016XS selected, initial configurations
   +    (/CS_n * CFG)           -> S2     ; 28F016XS selected, 28F016XS and i960 CA microprocessor configured
                               +-> S0     ; else, return to idle state
S2 :=   VCC                     -> S3
S3 :=   VCC                     -> S4     ; 28F016XS is configured to wait 4 clocks
S4 :=   (/BLAST_n * ADS_n)      -> S0     ; 1 double word  read
   +    (/BLAST_n * /ADS_n)     -> S1     ; pipelined read
                               +-> S5     ; else, continue
S5 :=   (/BLAST_n * ADS_n)      -> S0     ; burst read finished
  +     (/BLAST_n * /ADS_n)     -> S1     ; pipelined read
                               +-> S5     ; else, continue
S6 :=   /CS_n                   -> S7     ; 28F016XS selected, continue
                               +-> S0   ; else, return to idle state
S7 :=   VCC                     -> S8
S8 :=   (BLAST_n * CFG)         -> S6     ; continue burst
   +    (BLAST_n * /CFG)        -> S8     ; pre-config write
                               +-> S0     ; write is finished

; transition outputs
S0.OUTF :=      /OE * /WE
S1.OUTF :=      /OE * /WE
S2.OUTF :=       OE * /WE
S3.OUTF :=       OE * /WE
S4.OUTF :=       OE * /WE
S5.OUTF :=       OE * /WE
S6.OUTF :=       OE * /WE
S7.OUTF :=      /OE *  WE
S8.OUTF :=      /OE *  WE

EQUATIONS
  ; implement RESET
      GLOBAL.RSTF = /RESET
  ; implement 2-bit burst counter -  registered counter equations
      CTR1 := (LD * A3) + (/LD * INC * CTR0 * /CTR1)
            + (/LD * INC * /CTR0 * CTR1) + (/LD * /INC * CTR1)
      CTR0 := (LD * A2) + (/LD * INC * /CTR0) + (/LD * /INC * CTR0)
  ; flop OE and WE on falling edge
      OE.CLKF = /CLK
      WE.CLKF = /CLK
```

PLD file for the optimized 28F016XS interface to the i960 JF Microprocessor described in Section 3.0.

```
Title           Optimized 28F016XS/i960® JF Microprocessor Interface State Machine
Pattern         PDS
Revision        1
Authors         Example
Company         Intel Corporation - Folsom, California
Date            2-16-95

CHIP      STATEMACHINE      85C22V10

; inputs
PIN   1         CLK
PIN             ADS             ; address status - i960 JF microprocessor
PIN             W_R             ; W/R# - i960 JF microprocessor
PIN             BLAST           ; burst last - i960 JF microprocessor
PIN             CS              ; chip select
PIN             CFG             ; 28F016XS/i960 JF microprocessor config status set input
PIN             A2              ; A bit 2
PIN             A3              ; A bit 3
PIN             RESET           ; resets all FFs in device
PIN   25        GLOBAL          ; virtual pin to implement reset

; outputs
PIN             CTR0            ; burst counter out - 28F016XS-A1
PIN             CTR1            ; burst counter out - 28F016XS-A2
PIN             /WE             ; write enable - 28F016XS
PIN             /OE             ; output enable - 28F016XS
PIN             /RDYRCV         ; wait-state control
PIN             MUX             ; control data flow through the multiplexer
PIN             Q0              ; state variables
PIN             Q1
PIN             /ADV            ; state variable and address valid - 28F016XS
PIN             Q3

 ; burst counter control signals
STRING  LD '(/ADS)'             ; load
STRING  INC '(/ADV + /RDYRCV * Q3 * Q1 * Q0)'    ; increment

STATE MOORE_MACHINE
DEFAULT_BRANCH S0

; state assignments
          S0 =  /RDYRCV * /Q3 *  ADV * /Q1 * /Q0 * /OE * /WE * /MUX
          S1 =  /RDYRCV * /Q3 *  ADV * /Q1 *  Q0 *  OE * /WE* MUX
          S2 =  /RDYRCV * /Q3 *  ADV *  Q1 * /Q0 *  OE * /WE * MUX
          S3 =   RDYRCV * /Q3 *  ADV *  Q1 *  Q0 *  OE * /WE* MUX
          S4 =   RDYRCV * /Q3 * /ADV * /Q1 *  Q0 *  OE * /WE* MUX
          S5 =  /RDYRCV * /Q3 * /ADV *  Q1 * /Q0 * /OE * /WE* MUX
```

```
        S6 =  /RDYRCV *  Q3 *  ADV * /Q1 * /Q0 *  OE * /WE * MUX
        S7 =  /RDYRCV * /Q3 * /ADV *  Q1 *  Q0 *  OE * /WE* MUX
        S8 =  /RDYRCV *  Q3 *  ADV * /Q1 *  Q0 *  OE * /WE* MUX
        S9 =  RDYRCV *  Q3 * /ADV * /Q1 * /Q0 *  OE * /WE* MUX
        S10 = /RDYRCV *  Q3 *  ADV *  Q1 * /Q0 *  OE * /WE* MUX
        S11 = /RDYRCV *  Q3 * /ADV * /Q1 *  Q0 *  OE * /WE* MUX
        S12 = /RDYRCV *  Q3 * /ADV *  Q1 * /Q0 * /OE *  WE* MUX
        S13 = /RDYRCV *  Q3 * /ADV *  Q1 *  Q0 * /OE *  WE* MUX
        S14 =  RDYRCV *  Q3 * /ADV *  Q1 *  Q0 * /OE * /WE* MUX

; state transitions
S0 :=   (/ADS * /W_R *  CFG)  -> S1      ; READ cycle
  +     (/ADS * /W_R * /CFG) ->S5
  +     (/ADS *  W_R)          -> S13     ; WRITE cycle
                             +-> S0      ; else, stay
S1 :=    /CS                   -> S2      ; 28F016XS selected, init configurations
  +       CS                   -> S0      ; 28F016XS selected, optimized configured
S2 :=   VCC                    -> S3
S3 :=   (/BLAST)               -> S0      ; 1 double word read
                             +-> S4      ; else, continue
S4 :=   /BLAST                 -> S0      ; burst read finished
  +     (BLAST *  CFG)         -> S4      ; continue, optimized configuration
  +     (BLAST * /CFG)         -> S11     ; continue, initial configuration
S5 :=   VCC                    -> S6
S6 :=   VCC                    -> S7
S7 :=   VCC                    -> S8
S8 :=   VCC                    -> S9
S9 :=   /BLAST                 -> S0      ; BLAST -  end of the burst read transaction
  +     BLAST                  -> S10
S10:=   VCC                    -> S4
S11:=   VCC                    -> S4
S12:=   CS                     -> S0      ; write cycle control
  +     /CS                    -> S14
S13:=   VCC                    -> S15
S14:=   BLAST                  -> S13     ; BLAST - end of burst write transaction
  +     /BLAST                 -> S0

EQUATIONS
  ; implement RESET
      GLOBAL.RSTF = /RESET
  ; implement 2-bit burst counter -  registered counter equations
CTR1 := (/WR * LD * A3 * /A2) + (WR * LD * A3) + (/LD * INC * CTR0 * /CTR1)
       + (/LD * INC * /CTR0 * CTR1) + (/LD * /INC * CTR1)
CTR0 := (/WR * LD * /A2) + (WR * LD * A2) + (/LD * INC * /CTR0) + (/LD * /INC * CTR0)
```

```
PIN                 W_R                 ; W/R# - i960 JF microprocessor
PIN                 BLAST               ; burst last - i960 JF microprocessor
PIN                 CS                  ; chip select
PIN                 CFG                 ; 28F016XS/i960 JF microprocessor config status set input
PIN                 A2                  ; A bit 2
PIN                 A3                  ; A bit 3
PIN                 RESET               ; resets all FFs in device
PIN   25            GLOBAL              ; virtual pin to implement reset

; outputs
PIN                 CTR0                ; burst counter out - 28F016XS-A1
PIN                 CTR1                ; burst counter out - 28F016XS-A2
PIN                 /WE                 ; write enable - 28F016XS
PIN                 /OE                 ; output enable - 28F016XS
PIN                 /RDYRCV             ; wait-state control
PIN                 Q0                  ; state variables
PIN                 Q1
PIN                 /ADV                ; state variable and address valid - 28F016XS
PIN                 Q3

 ; burst counter control signals
STRING  LD '(/ADS)'                     ; load
STRING  INC '(/ADV + /RDYRCV * Q3 * Q1 * Q0)'    ; increment

STATE MOORE_MACHINE
DEFAULT_BRANCH S0

; state assignments
        S0 =    /RDYRCV * /Q3 * /ADV * /Q1 * /Q0 * /OE * /WE
        S1 =    /RDYRCV * /Q3 *  ADV * /Q1 * /Q0 * /OE * /WE
        S2 =    /RDYRCV * /Q3 *  ADV * /Q1 *  Q0 *  OE * /WE
        S3 =    /RDYRCV * /Q3 *  ADV *  Q1 * /Q0 *  OE * /WE
        S4 =     RDYRCV * /Q3 *  ADV *  Q1 *  Q0 *  OE * /WE
        S5 =     RDYRCV * /Q3 * /ADV * /Q1 *  Q0 *  OE * /WE
        S6 =    /RDYRCV * /Q3 * /ADV *  Q1 * /Q0 * /OE * /WE
```

```
        S7 =  /RDYRCV *  Q3 *  ADV * /Q1 * /Q0 *  OE * /WE
        S8 =  /RDYRCV * /Q3 * /ADV *  Q1 *  Q0 *  OE * /WE
        S9 =  /RDYRCV *  Q3 *  ADV * /Q1 *  Q0 *  OE * /WE
        S10 =  RDYRCV *  Q3 * /ADV * /Q1 * /Q0 *  OE * /WE
        S11 = /RDYRCV *  Q3 *  ADV *  Q1 * /Q0 *  OE * /WE
        S12 = /RDYRCV *  Q3 * /ADV * /Q1 *  Q0 *  OE * /WE
        S13 = /RDYRCV *  Q3 * /ADV *  Q1 * /Q0 * /OE *  WE
        S14 = /RDYRCV *  Q3 * /ADV *  Q1 *  Q0 * /OE *  WE
        S15 =  RDYRCV *  Q3 * /ADV *  Q1 *  Q0 * /OE * /WE

; state transitions
S0 :=   (/ADS * /W_R)          -> S1     ; READ cycle
  +     (/ADS *  W_R)          -> S13    ; WRITE cycle
                              +-> S0     ; else, stay
S1 :=   (/CS * /CFG)           -> S6     ; 28F016XS selected, init configurations
  +     (/CS * CFG)            -> S2     ; 28F016XS selected, optimized configured
                              +-> S0     ; else, return to idle state
S2 :=   VCC                    -> S3
S3 :=   VCC                    -> S4     ; 28F016XS is configured to wait 4 clocks
S4 :=   (/BLAST * ADS)         -> S0     ; 1 double word read
                              +-> S5     ; else, continue
S5 :=   /BLAST                 -> S0     ; burst read finished
  +     (BLAST *  CFG)         -> S5     ; continue, optimized configuration
  +     (BLAST * /CFG)         -> S12    ; continue, initial configuration
S6 :=   VCC                    -> S7
S7 :=   VCC                    -> S8
S8 :=   VCC                    -> S9
S9 :=   VCC                    -> S10
S10:=   /BLAST                 -> S0     ; BLAST -  end of the burst read transaction
  +     BLAST                  -> S11
S11:=   VCC                    -> S5
S12:=   VCC                    -> S5
S13:=   CS                     -> S0     ; write cycle control
  +     /CS                    -> S14
S14:=   VCC                    -> S15
S15:=   BLAST                  -> S13    ; BLAST - end of burst write transaction
  +     /BLAST                 -> S0

EQUATIONS
  ; implement RESET
      GLOBAL.RSTF = /RESET
  ; implement 2-bit burst counter -  registered counter equations
      CTR1 := (LD * A3) + (/LD * INC * CTR0 * /CTR1)
            + (/LD * INC * /CTR0 * CTR1) + (/LD * /INC * CTR1)
      CTR0 := (LD * A2) + (/LD * INC * /CTR0) + (/LD * /INC * CTR0)
```

APPENDIX B
BENCHMARK PERFORMANCE ANALYSIS

The following section provides detailed memory technology information used in the performance analysis (UDP/IP Networking and Imaging Benchmarks) contained in the introduction. The performance analysis was based on actual memory component performance in an i960 processor-based environment. System interface delay between microprocessor and memory was not included in the analysis. The two benchmarks illustrate relative system memory performance.

A. 28F016XS Flash Memory

28F016XS is capable of 3-1-1-1-1-1-1-1. . .read performance at 5.0V V_{CC} and 33 MHz or 25 MHz (2-0-0-0-0-0-0-0. . .in terms of wait-states). The benchmarking analysis is shown below:

	UDP/IP Networking Benchmark	Imaging Benchmark
	Time (sec)	*Time (sec)*
i960 KB-25 Microprocessor	1.30	1.64
i960 CA-33 Microprocessor	.89	.89
i960 CF-33 Microprocessor	.53	.59

B. 16-Mbit DRAM

16-Mbit DRAMs were, at the time this technical paper was published, only beginning to ramp into production. Only advance information for the wider x16, 16-Mbit DRAMs was available for use in the calculations that follow.

Sequential reads allow use of the DRAM fast page mode. Assumed DRAM specifications are shown below:

- 80 ns t_{RAC}, 40 ns t_{AA} (5.0V V_{CC})
- 256 word (512 byte) page buffer

Therefore, 16-Mbit DRAMs are capable of 3-2-2-2-2-2-2-2. . .read performance at 33 MHz and 25 MHz (2-1-1-1-1-1-1-1 in terms of wait-states. . .). The benchmarking analysis is shown below:

	UDP/IP Networking Benchmark	Imaging Benchmark
	Time (sec)	*Time (sec)*
i960 KB-25 Microprocessor	1.88	1.89
i960 CA-33 Microprocessor	1.06	1.03
i960 CF-33 Microprocessor	.59	.64

i960 KB-25 Microprocessor	NA	NA
i960 CA-33 Microprocessor	1.56	1.49
i960 CF-33 Microprocessor	.78	.81

D. 16-Mbit PAGED MASK ROM

Calculations that follow used the x16 version of the 16-Mbit paged mask ROM, which is not yet widely available from multiple vendors. The x8, 16-Mbit paged mask ROM is the more common version today.

Sequential reads allow use of the mask ROM page mode. The assumed 5.0V V_{CC} 16-Mbit mask ROM random access time is 150 ns, with 75 ns accesses in page mode (4-word page). Therefore, 16-Mbit mask ROMs are capable of 5-3-3-3-5-3-3-3. . .read performance at 5.0V V_{CC} and 33 MHz (4-2-2-2-4-2-2-2. . .in terms of wait-states). The benchmarking analysis is shown below:

	UDP/IP Networking Benchmark	Imaging Benchmark (sec)
	Time (sec)	*Time (sec)*
i960 KB-25 Microprocessor	NA	NA
i960 CA-33 Microprocessor	1.35	1.30
i960 CF-33 Microprocessor	.71	.73

ADVANCE INFORMATION

VALUE SERIES 100

FLASH MEMORY CARD 2-,4-,8-MEGABYTE

iMC002FLSC, iMC004FLSC, iMC008FLSC

- **Low-Cost Linear Flash Card**
- **Single Supply: 5 Volt Operation**
- **FAST Read Performance — 100 ns Maximum Access Time**
- **x16 Data Interface**
- **High-Performance Random Writes — 10 µs Typical Word Write**
- **50 µA Typical Deep Power-Down**
- **Automated Write and Erase Algorithms — 28F008SA Command Set**
- **State-of-the-Art 0.6 µm ETOX™ IV Flash Technology**
- **100,000 Erase Cycles per Block**
- **64-Kword Blocks**
- **PC Card Standard Type 1 Form Factor**

Intel's Value Series 100 card offers the lowest cost removable solid-state storage solution for code and data storage, high-performance disk emulation, and applications in mobile PCs and dedicated embedded applications. Manufactured with Intel's FlashFile™ memory, this card takes advantage of a revolutionary architecture that provides innovative capabilities, automated write/erase algorithms, reliable operation and very high read/write performance.

The flash memory card provides the lowest cost, highest performance nonvolatile read/write solution for solid-state storage applications. These applications are enhanced further with this product's symmetrically-blocked architecture, extended MTBF, and 5 Volt operation.

The flash memory card can be used as a simple x16 linear array of flash devices. The PC Card form factor offers an industry-standard pinout, removable linear flash memory, and the ability to upgrade system memory software without changing the board layout.

Order Number: 290546-001

intel

ADVANCE INFORMATION

VALUE SERIES 100

FLASH MEMORY CARD

2-,4-,8-MEGABYTE

MC002FLSC, MC004FLSC, MC008FLSC

- Low-Cost Linear Flash Card
- Single Supply: 5 Volt Operation
- FAST Read Performance
  - 100 ns Maximum Access Time
- x16 Data Interface
- High-Performance Random Writes
  - 10 µs Typical Word Write
- 50 µA Typical Deep Power-Down
- Automated Write and Erase Algorithms
  - 28F008SA Command Set
- State-of-the-Art 0.6 µm ETOX™ IV Flash Technology
- 100,000 Erase Cycles per Block
- 64-Kword Blocks
- PC Card Standard Type 1 Form Factor

Intel's Value Series 100 card offers the lowest cost removable solid-state storage solution for code and data storage. High-performance disk emulation, and applications in mobile PCs and dedicated embedded applications. Manufactured with Intel's FlashFile™ memory, this card takes advantage of a revolutionary architecture that provides innovative capabilities, automated write/erase algorithms, reliable operation and very high read/write performance.

The flash memory card provides the lowest cost, highest performance nonvolatile read/write solution for solid-state storage applications. These applications are enhanced further with this product's symmetrically-blocked architecture, extended MTBF, and 5 Volt operation.

The flash memory card can be used as a simple x16 linear array of flash devices. The PC Card form factor offers an industry-standard pinout, removable linear flash memory, and the ability to upgrade system memory software without changing the board layout.

Order Number 290542-001

1.0 SCOPE OF DOCUMENT

This datasheet provides a card architecture overview, all AC and DC characteristics and command definitions.

2.0 PRODUCT OVERVIEW

The 2-, 4-, and 8-megabyte flash memory cards each contain a flash memory array that consists of two, four, or eight 28F008SA (1-Mbyte) TSOP memory devices, respectively. Each 28F008SA contains 16 distinct, individually-erasable, 64-Kbyte blocks. Therefore, the 2-, 4-, and 8-megabyte cards contain 32, 64 and 128 independently-erasable blocks.

At the device level, internal algorithm automation allows execution of write and erase operations using a two-write command sequence in the same way as the 28F008SA FlashFile™ memory in the Series 2 card. The automated write/erase algorithms ensure that data is reliably written in the least amount of time.

The memory card interface supports the PC Card Standard, supported by Personal Computer Memory Card Industry Association (PCMCIA) and Japanese Electronics Industry Development Association (JEIDA) 68-pin card format. The Value Series 100 card meets all PCMCIA/JEIDA Type 1 mechanical specifications.

3.0 VALUE SERIES 100 CARD ARCHITECTURE OVERVIEW

The Value Series 100 is a simple array of flash devices in a PC Card form factor, offering the lowest cost, removable memory solution at 2-, 4- and 8-MB densities. Two 28F008SA devices in parallel provide the lower and upper bytes for a 16-bit access. The on-board charge pump provides 12.0V to the flash array at all times for write and erase operations in an environment where only a 5V supply is available. The Card Information Structure (CIS) for the Value Series 100 card is stored in Block 0 of the flash memory to reduce the attribute memory cost overhead of an EEPROM or ASIC. In embedded applications, a CIS may not be required by the system and the entire memory array can be used by the system.

3.1 Card Pinout and Pin Description

The 68-pin PC Card format provides the system interface for the Value Series 100 card (see Tables 1 and 2). The detailed specifications for this interface is described in the PC Card Standard Specification. The Value Series 100 card product family conforms to the pinout requirements of previous PCMCIA Versions Release 1.0, Release 2.0 and Release 2.01 and the PC Card Standard.

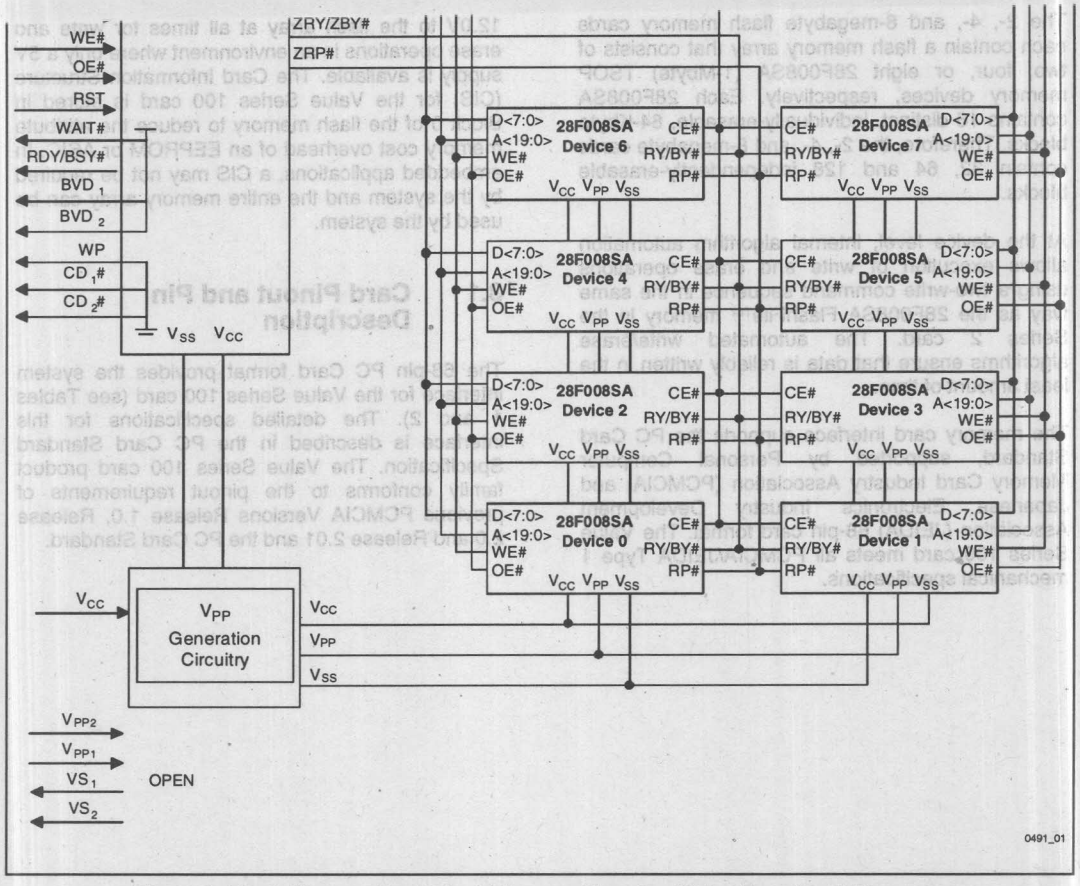


Figure 1. Flash Memory Card Block Diagram Showing Major Functional Elements

Table 1. Value Series 100 Card Signals

Pin	Signal	I/O	Function	Active
1	GND		Ground	
2	DQ_3	I/O	Data Bit 3	
3	DQ_4	I/O	Data Bit 4	
4	DQ_5	I/O	Data Bit 5	
5	DQ_6	I/O	Data Bit 6	
6	DQ_7	I/O	Data Bit 7	
7	CE_1#	I	Card Enable 1	LOW
8	A_{10}	I	Address Bit 10	
9	OE#	I	Output Enable	LOW
10	A_{11}	I	Address Bit 11	
11	A_9	I	Address Bit 9	
12	A_8	I	Address Bit 8	
13	A_{13}	I	Address Bit 13	
14	A_{14}	I	Address Bit 14	
15	WE#	I	Write Enable	LOW
16	RDY/BSY#	O	Ready/Busy	LOW
17	V_{CC}		Supply Voltage	
18	V_{PP1}		Supply Voltage	N.C.
19	A_{16}	I	Address Bit 16	
20	A_{15}	I	Address Bit 15	
21	A_{12}	I	Address Bit 12	
22	A_7	I	Address Bit 7	
23	A_6	I	Address Bit 6	
24	A_5	I	Address Bit 5	
25	A_4	I	Address Bit 4	
26	A_3	I	Address Bit 3	

Pin	Signal	I/O	Function	Active
27	A_2	I	Address Bit 2	
28	A_1	I	Address Bit 1	
29	A_0	I	Address Bit 0	
30	DQ_0	I/O	Data Bit 0	
31	DQ_1	I/O	Data Bit 1	
32	DQ_2	I/O	Data Bit 2	
33	WP	O	Write Protect	HIGH
34	GND		Ground	
35	GND		Ground	
36	CD_1#	O	Card Detect 1	LOW
37	DQ_{11}	I/O	Data Bit 11	
38	DQ_{12}	I/O	Data Bit 12	
39	DQ_{13}	I/O	Data Bit 13	
40	DQ_{14}	I/O	Data Bit 14	
41	DQ_{15}	I/O	Data Bit 15	
42	CE_2#	I	Card Enable 2	LOW
43	VS_1	O	Voltage Sense 1	N.C.
44	RFU		Reserved	
45	RFU		Reserved	
46	A_{17}	I	Address Bit 17	
47	A_{18}	I	Address Bit 18	
48	A_{19}	I	Address Bit 19	
49	A_{20}	I	Address Bit 20	
50	A_{21}	I	Address Bit 21	
51	V_{CC}		Supply Voltage	
52	V_{PP2}		Supply Voltage	N.C.

Table 1. Value Series 100 Card Signals (Continued)

Pin	Signal	I/O	Function	Active
53	A_{22}	I	Address Bit 22	
54	A_{23}	I	Address Bit 23	
55	A_{24}	I	Address Bit 24	N.C.
56	A_{25}	I	Address Bit 25	N.C.
57	VS_2	O	Voltage Sense 2	N.C.
58	RST	I	Reset	HIGH
59	WAIT#	O	Extend Bus Cycle	LOW
60	RFU		Reserved	

Pin	Signal	I/O	Function	Active
61	REG#	I	Attribute Memory Select	
62	BVD_2	O	Battery Voltage Detect 2	
63	BVD_1	O	Battery Voltage Detect 1	
64	DQ_8	I/O	Data Bit 8	
65	DQ_9	I/O	Data Bit 9	
66	DQ_{10}	I/O	Data Bit 10	
67	CD_2#	O	Card Detect 2	LOW
68	GND		Ground	

Table 2. Value Series 100 Card Signal Description

Symbol	Type	Name and Function
A_0–A_{25}	INPUT	**ADDRESS INPUTS:** Address A_0 through A_{25} are address bus lines which enable direct addressing of up to 64 MB of memory on the card. Signal A_0 is not decoded since the card is x16 only. The memory will wrap at the card density boundary. The system should NOT try to access memory beyond the card's density, since the upper addresses are not decoded.
DQ_0–DQ_{15}	INPUT/ OUTPUT	**DATA INPUT/OUTPUT:** DQ_0 through DQ_{15} constitute the bi-directional data bus. DQ_{15} is the most significant bit.
CE_1#, CE_2#	INPUT	**CARD ENABLE 1 & 2:** CE_1# enables EVEN byte accesses on D_{0-7}, CE_2# enables ODD byte accesses on D_{8-15}. Cannot access Odd Bytes on D_{0-7}.
OE#	INPUT	**OUTPUT ENABLE:** Active low signal gating read data from the memory card.
WE#	INPUT	**WRITE ENABLE:** Active low signal gating write data to the memory card.
RDY/BSY#	OUTPUT	**READY/BUSY OUTPUT:** Indicates status of internally timed erase or write activities. A high output indicates the memory card is ready to accept accesses.
CD_1#, CD_2#	OUTPUT	**CARD DETECT 1 & 2:** These signals provide for card insertion detection. The signals are connected to ground internally on the memory card, and will be forced low whenever a card is placed in the socket. The host socket interface circuitry shall supply 10K or larger pull-up resistors on these signal pins.
WP	OUTPUT	**WRITE PROTECT:** This signal is pulled LOW for PC Card Standard compatibility. The flash memory card has no WP signal functionality.
V_{PP1},V_{PP2}	N.C.	**WRITE/ERASE POWER SUPPLY:** These power signals are not connected for the 5V-only card.
V_{CC}		**CARD POWER SUPPLY:** 5.0V for all internal circuitry.
GND		**GROUND** for all internal circuitry.
REG#	INPUT	**REGISTER SELECT:** The memory card has no separate attribute memory. The CIS is located in common memory. REG# is unconnected on the card.
RST	INPUT	**RESET:** Active high signal for placing card in Power-On Default State. RESET can be used as a POWER-DOWN signal for the memory array.
WAIT#	OUTPUT	**WAIT:** (Extended Bus Cycle) This signal is pulled high for compatibility.
BVD_1, BVD_2	OUTPUT	**BATTERY VOLTAGE DETECT:** These signals are pulled high to maintain SRAM card compatibility.
VS_1, VS_2	OUTPUT	**VOLTAGE SENSE:** Notifies the host socket of the card's V_{CC} requirements. VS_1 and VS_2 are OPEN to indicate a 5V, 16-bit card has been inserted.
RFU		**RESERVED FOR FUTURE USE**
N.C.		**NO INTERNAL CONNECTION TO CARD** pin may be driven or left floating.

The components on the Value Series 100 card have three read modes: read memory array, read intelligent identifier or read Status Register.

Read modes are enabled by writing the appropriate read mode command to the Command User Interface (CUI). The 28F008SA automatically resets to read array mode upon initial device power-up, or after reset. The 28F008SA has four control pins, two of which must be logically active to obtain data at the outputs. Chip Enables ($CE_{1,2}$#) are the device selection control, and, when active, enable the selected memory device. Output Enable (OE#) is the data input/output (DQ_0–DQ_{15}) direction control, and, when active, drives data from the selected memory onto the I/O bus. WE# must be driven to V_{IH} during a read access.

4.1.2 OUTPUT DISABLE

With OE# and WE# at a logic-high level (V_{IH}), the device outputs are disabled. Output (DQ_0–DQ_{15}) are placed in a high-impedance state.

4.1.3 STANDBY

$CE_{1,2}$# at a logic-high level (V_{IH}) places the card in standby mode. Standby operation disables much of the card's circuitry and substantially reduces device power consumption. The outputs (DQ_0–DQ_{15}) are placed in a high-impedance state independent of the status of OE#. If the card is de-selected during byte write or block erase, the card will continue functioning and consuming normal active power until the operation completes.

location 00000H outputs the manufacturer code (8989H). A read from address 00002H outputs the device code: A2A2H, A6A6H or AAAAH.

4.1.5 WRITE

Writes to the CUI enable reading of device data and intelligent identifiers. They also control inspection and clearing of the Status Register. The contents of the interface register serves as input to the internal state machine on each component.

The CUI itself does not occupy an addressable memory location. The interface register is a latch used to store the command, address and data information needed to execute the command. Erase Setup and Erase Confirm commands require both appropriate command data and an address within the block to be erased. The Write Setup command requires both appropriate command data and the address of the location to be written, while the Write command consists of the data to be written and the address of the location to be written.

The CUI is written by bringing WE# to a logic-low level (V_{IL}) while CE# is low. Addresses and data are latched on the rising edge of WE#. Standard microprocessor write timings are used.

Refer to AC Write Characteristics and the AC Waveforms for Write Operations, Figure 14, for specific timing parameters.

4.2 Address Decode Logic

The address decode logic selects which components device pair is enabled during a read or write access. Unused upper addresses for the Value Series 100 card will not be decoded. The address decoding will wrap around at the card's density.

The Value Series 100 card does not have a separate attribute memory space and REG# is not included in the address decode logic. REG# accesses will result in a read/write to the common memory flash array.

4.3 Data Control

As shown in Table 3, data paths and directions are selected by the Data Control logic using WE#, OE#, CE_1#, and CE_2# as logic inputs. The Data Control logic selects any of the PCMCIA word-wide, EVEN-byte and ODD-byte modes for either reads or writes to common memory.

NOTE:

This card has a x16 interface. The ODD byte CANNOT be accessed on the lower data path (D_{0-7}). A_0 is not decoded.

Table 3. Data Access Mode Truth Table

Mode	RESET	CE_2#	CE_1#	OE#	WE#	A_1	V_{PP}	D_{8-15}	D_{0-7}	RY/BY#	Notes
Even Byte-Read	V_{IL}	V_{IH}	V_{IL}	V_{IL}	V_{IH}	X	X	High-Z	Even	X	1,2,3
Odd Byte-Read	V_{IL}	V_{IL}	V_{IH}	V_{IL}	V_{IH}	X	X	Odd	High-Z	X	1,2,3
Word-Read	V_{IL}	V_{IL}	V_{IL}	V_{IL}	V_{IH}	X	X	Odd	Even	X	1,2,3
Even Byte-Write	V_{IL}	V_{IH}	V_{IL}	V_{IH}	V_{IL}	X	X	XXX	Even	V_{OL}	3,4
Odd Byte-Write	V_{IL}	V_{IL}	V_{IH}	V_{IH}	V_{IL}	X	X	Odd	XXX	V_{OL}	3,4
Word-Write	V_{IL}	V_{IL}	V_{IL}	V_{IH}	V_{IL}	X	X	Odd	Even	V_{OL}	3,4
Manufacturer ID	V_{IL}	V_{IL}	V_{IL}	V_{IL}	V_{IH}	V_{IL}	X	89H	89H	V_{OH}	
Device ID	V_{IL}	V_{IL}	V_{IL}	V_{IL}	V_{IH}	V_{IH}	X	A2H	A2H	V_{OH}	5
Standby	V_{IL}	V_{IH}	V_{IH}	X	X	X	X	High-Z	High-Z	X	3
Output Disable	V_{IL}	X	X	V_{IH}	V_{IH}	X	X	High-Z	High-Z	X	3
Power-Down	V_{IH}	X	X	X	X	X	X	High-Z	High-Z	X	

NOTES:

1. Refer to DC Characteristics.
2. X can be V_{IL} or V_{IH} for control pins and address.
3. RDY/BSY# is V_{OL} when the WSM is executing internal byte write or block erase algorithms. It is V_{OH} when the WSM is not busy, in erase suspend mode, or deep power-down mode.
4. Refer to Table 4 for valid D_{IN} during a write operation.
5. The device code can be A2H, A6H or AAH. Software should check for all three cases for compatibility with future cards.

5.0 COMMAND DEFINITIONS

Read operations from the Status Register, intelligent identifiers or array blocks are always enabled.

Device operations are selected by writing specific commands into the Command User Interface. Table 4 defines the 28F008SA commands.

5.1 Read Array Command (FFFFH)

Upon initial device power-up, and after reset, the 28F008SA defaults to read array mode. This operation is also initiated by writing FFFFH into the CUI on the component. Microprocessor read cycles retrieve array data. The device remains enabled for reads until the CUI contents are altered by issuing a Write command. Once the internal WSM has started a byte write or block erase operation, the device will not recognize the Read Array command until the WSM has completed its operation.

5.2 Intelligent Identifier Command (9090H)

The 28F008SA contains an intelligent identifier operation, initiated by writing 9090H into the CUI. Following the command write, a ready cycle from address 00000H retrieves the manufacturer code of 8989H. A read cycle from address 00002H returns the device code of A2A2H, A6A6H or AAAAH. Software should recognize To terminate the operation, it is necessary to write another valid command into the register.

Table 4. 28F008SA-Compatible Mode Command Bus Definitions

Command	First Bus Cycle				Second Bus Cycle			
	R/W	Addr	Data		R/W	Addr	Data	
			Byte	Word			Byte	Word
Read Array	W	DA	FFH	FFFFH	R	DA	AD	AD
Intelligent Identifier	W	DA	90H	9090H	R	IA	ID	ID
Read Status Register	W	DA	70H	7070H	R	DA	CSRD	CSRD
Clear Status Register	W	DA	50H	5050H				
Word/Byte Write	W	WA	40H	4040H	W	WA	WD	WD
Word/Byte Write (Alternate)	W	WA	10H	1010H	W	WA	WD	WD
Block Erase/Confirm	W	BA	20H	2020H	W	BA	D0H	D0D0H
Erase Suspend/Resume	W	DA	B0H	B0B0H	W	DA	D0H	D0D0H

ADDRESSES:

DA	Device Address
BA	Block Address
IA	Identifier Address
WA	Write Address

DATA:

AD	Array Data
CSRD	Status Reg. Data
ID	Identifier Data
WD	Write Data

5.3 Read Status Register Command (7070H)

The 28F008SA components on the Value Series 100 card each contain a Status Register which may be read to determine when a write or block erase operation is complete, and whether that operation completed successfully. The Status Register may be read at any time by writing the Read Status Register command (7070H) to the CUI. After writing this command, all subsequent read operations output data from the Status Register, until another valid command is written to the CUI. The contents of the Status Register are latched on the falling edge of OE# or CE#, whichever occurs first. OE# or CE# must be toggled to V_{IH} before further reads to update the Status Register latch.

NOTE: Two 28F008SA devices are used in parallel to form a x16 operation. Both status registers need to be checked when determining the status of a x16 Erase/Write operation.

5.4 Clear Status Register Command (5050H)

The Erase Status and Write Status bits are set to "1"s by the WSM and can only be reset by the Clear Status Register command. These bits indicate various failure conditions (see Table 5). By allowing system software to control the resetting of these bits, several operations may be performed (such as cumulatively writing several bytes or erasing multiple blocks in sequence). The Status Register may then be polled to determine if an error occurred during that sequence. This Status Register functionality adds flexibility to the way the device may be used.

Additionally, the V_{PP} Status bit (SR.3) MUST be reset by system software before further writes or block erases are attempted. To clear the Status Register, the Clear Status Register command (50H) is written to the CUI.

Table 5. Status Register Definition

WSMS	ESS	ES	BWS	VPPS	R	R	R
7	6	5	4	3	2	1	0

SR.7 = WRITE STATE MACHINE STATUS
1 = Ready
0 = Busy

SR.6 = ERASE SUSPEND STATUS
1 = Erase Suspended
0 = Erase in Progress/Completed

SR.5 = ERASE STATUS
1 = Error in Block Erasure
0 = Successful Block Erase

SR.4 = BYTE WRITE STATUS
1 = Error in Byte Write
0 = Successful Byte Write

SR.3 = V_{PP} STATUS
1 = V_{PP} Low Detect, Operation Abort
0 = V_{PP} OK

SR.2-SR.0 = RESERVED FOR FUTURE ENHANCEMENTS.

NOTES:

RDY/BSY# or the Write State Machine Status bit must first be checked to determine byte write or block erase completion, before the Byte Write or Erase Status bit are checked for success.

If the Byte Write *and* Erase Status bits are set to "1"s during a block erase attempt, an improper command sequence was entered. Attempt the operation again.

If V_{PP} low status is detected, the Status Register must be cleared before another byte write or block erase operation is attempted.

The V_{PP} Status bit, unlike an A/D converter, does not provide continuous indication of V_{PP} level. The WSM interrogates the V_{PP} level only after the Write or Block Erase command sequence have been entered and informs the system if V_{PP} has not been switched on.

These bits are reserved for future use and should be masked out when polling the Status Register

handled internally by the WSM, invisible to the system. After the two-command erase sequence is written to it, the 28F008SA automatically outputs Status Register data when read (see Figure 8, Block Erase Flowchart). The CPU can detect the completion of the erase event by analyzing the output data of the RDY/BSY# pin, or the WSM Status bit of the Status Register. When erase is completed, the Erase Status bit should be checked. If erase error is detected the Status Register should be cleared. The CUI remains in read Status Register mode until further commands are issued to it.

5.7 Erase Suspend/Erase Resume Commands (B0B0H,D0D0H)

The Erase Suspend command allows block erase interruption in order to read data from another block of memory. Once the erase process starts, writing the Erase Suspend command (B0B0H) to the CUI requests that the WSM suspend the erase sequences at a predetermined point in the erase algorithm. The 28F008SA continues to output Status Register data when read, after the Erase Suspend command is written to it. Polling the WSM Status and Erase Suspend Status bits will determine when the erase operation has been suspended (both will be set to "1"). RDY/BSY# will also transition to V_{OH}.

At this point, a Read Array command can be written to the CUI to read data from blocks other than that which is suspended. The only other valid commands, at this time, are Read Status Register (7070H) and Erase Resume (D0D0H), at which time the WSM will continue with the erase process. The Erase Suspend Status and WSM Status bits of the Status Register will automatically cleared and the

sequence. The Write Setup command (4040H or 1010H) is written to the CUI, followed by a second write specifying the address and data (latched on the rising edge of WE#) to be written. The WSM then takes over, controlling the byte write and write verify algorithms internally. After the two-command write sequence is written to it, the 28F008SA automatically outputs Status Register data when read (see Figure 10, Write Flowchart). The CPU can detect the completion of the byte write event by analyzing the output of the RDY/BSY# pin, or the WSM Status bit of the Status Register. Only the Read Status Register command is valid while write is active.

When write is complete, the Write Status bit should be checked. If byte write error is detected, the Status Register should be cleared. The internal WSM verify only detects errors for "1"s that do not successfully write to "0"s. The CUI remains in read Status Register mode until further commands are issued to it.

6.0 PC CARD INFORMATION STRUCTURE

The Card Information Structure (CIS) begins at address 00000000H of the card's Common Memory Plane in the EVEN byte locations. It contains a variable length chain of data blocks (tuples) that conform to a basic format (Table 6). The CIS of the Value Series 100 card is found in Table 7.

CAUTION:

The CIS data in Block 0 is not write protected and should not be erased by the system software if the CIS is needed for card recognition.

Table 6. PC Card Tuple Format

Bytes	Data
0	Tuple Code: CISTPL_xxx. The tuple code 0FFH indicates no more tuples in the list.
1	Tuple Link: TPL_LINK. Link to the next tuple in the list. This can be viewed as the number of additional bytes in tuple, excluding this byte. A link field of zero indicates an empty tuple body. A link field containing 0FFH indicates the last tuple in the list.
2-n	Bytes specific to this tuple.

Table 7. Value Series 100 Card Tuples

Address	Value	Description
00H	01H	**CISTPL_DEVICE**
02H	03H	TPL_LINK
04H	54H	FLASH = 100 ns
06H	06H	CARD SIZE: 2 MB
	0EH	4 MB
	1EH	8 MB
08H	FFH	END OF DEVICE
0AH	1EH	**CISTPL DEVICEGEO**
0CH	06H	TPL_LINK
0EH	02H	DGTPL_BUS
10H	11H	DGTPL_EBS
12H	01H	DGTPL_RBS
14H	01H	DGTPL_WBS
16H	03H	DGTPL_PART = 1
18H	01H	FLASH DEVICE INTERLEAVE
1AH	20H	**CISTPL_MANFID**
1CH	04H	TPL_LINK (04H)
1EH	89H	TPLMID_MANF: LSB
20H	00H	TPLMID_MANF: MSB
22H	03H	2 MB - 100 ns
	13H	4 MB - 100 ns
	23H	8 MB - 100 ns
24H	85H	TPLMID_CARD MSB
26H	21H	**CISTPL_FUNCID**
28H	02H	TPL_LINK
2AH	01H	TPLFID_FUNCTION : Memory
2CH	00H	TPLFID_SYSINIT
2EH	12H	**CISTPL_LONGLINK_C**
30H	04H	TPL_LINK
32H	00H	LOWEST BYTE
34H	00H	
36H	02H	
38H	00H	HIGHEST BYTE
3AH	15H	**CISTPL_VERS1**
3CH	40H	TPL_LINK
3EH	05H	TPLLV1_MAJOR
40H	00H	TPLLV1_MINOR
42H	69H	TPLLV1_INFO i
44H	6EH	n
46H	74H	t
48H	65H	e
4AH	6CH	l
4CH	00H	END TEXT
4EH	56H	V
50H	41H	A
52H	4CH	L
54H	55H	U
56H	45H	E

Address	Value	Description
58H	20H	SPACE
5AH	53H	S
5CH	45H	E
5EH	52H	R
60H	49H	I
62H	45H	E
64H	53H	S
66H	20H	SPACE
68H	31H	1
6AH	30H	0
6CH	30H	0
6EH	20H	SPACE
70H	00H	END TEXT
72H	30H 30H 30H	2MB 4MB 8MB
74H	32H 34H 38H	2MB 4MB 8MB
76H	20H	SPACE
78H	00H	END TEXT
7AH	43H	C
7CH	4FH	O
7EH	50H	P
80H	59H	Y
82H	52H	R
84H	49H	I
86H	47H	G
88H	48H	H
8AH	54H	T
8CH	20H	SPACE

Address	Value	Description
8EH	49H	I
90H	4EH	N
92H	54H	T
94H	45H	E
96H	4CH	L
98H	20H	SPACE
9AH	43H	C
9CH	4FH	O
9EH	52H	R
A0H	50H	P
A2H	4FH	O
A4H	52H	R
A6H	41H	A
A8H	54H	T
AAH	49H	I
ACH	4FH	O
AEH	4EH	N
B0H	20H	SPACE
B2H	31H	1
B4H	39H	9
B6H	39H	9
B8H	35H	5
BAH	00H	END TEXT
BCH	FFH	END OF LIST
BEH	FFH	**CISTPL_END**
C0H	00H	INVALID ADDRESS

7.0 SYSTEM DESIGN CONSIDERATIONS

7.1 Power Supply Decoupling

Flash memory power-switching characteristics require careful device decoupling. System designers are interested in three supply current issues: standby, active and transient current peaks which are produced by rising and falling edges of CE_1# and CE_2#. The capacitive and inductive loads on the card and internal flash memory device pairs determine the magnitudes of these peaks.

Three-line control and proper decoupling capacitor selection suppress transient voltage peaks. The Value Series 100 cards contain on-card ceramic decoupling capacitors connected between V_{CC} and GND.

The card connector should also have a 4.7 µF electrolytic capacitor between V_{CC} and GND. The bulk capacitors overcome voltage slumps caused by printed-circuit-board trace inductance, and supply charge to the smaller capacitors as needed.

7.2 Power-Up/Down Protection

The PCMCIA/JEIDA-specified socket properly sequences the power supplies to the flash memory card via shorter and longer pins.

Each device in the memory card is designed to offer protection against accidental erasure or writing, caused by spurious system-level signals that may exist during power transitions. The card will power-up into the read state.

A system designer must guard against active writes for V_{CC} voltages above V_{LKO} (2.0V). Since both WE# and CE_1# must be low for a command write, driving either to V_{IH} will inhibit writes. With its control register architecture, alteration of device contents only occurs after successful completion of the two-step command sequences.

7.3 RDY/BSY# and Byte Write/Block Erase Polling

RDY/BSY# is a full CMOS output that provides a hardware method of detecting byte write and block erase completion. It transitions low time t_{WHRL} after a Write or Erase command sequence is written to a 28F008SA, and returns to V_{OH} when all the WSM has finished executing the internal algorithm.

RDY/BSY# can be connected to the interrupt input of the system CPU or controller. It is active at all times. RDY/BSY# is also V_{OH} when the device is in erase suspend or deep power-down modes.

7.4 V_{CC}, V_{PP}, RESET Transitions and the Command/Status Registers

Write and block erase completion are not guaranteed if the internally generated V_{PP} drops below V_{PPH}. If the V_{PP} Status bit of the Status Register (SR.3) is set to "1," a Clear Status Register command *must* be issued before further write/block erase attempts are allowed by the WSM. Otherwise, the Write (SR.4) or Erase Status (SR.5) bits of the Status Register will be set to "1"s, if error is detected. RESET transitions to V_{IH} during write and block erase also abort the operations. Data is partially altered in either case, and the command sequence must be repeated after normal operation is restored. Device power-off, or RESET transitions to V_{IH}, clear the Status Register to initial value 10000XXX for the upper eight bits.

The CUI latches commands, as issued by system software, and is not altered by CE# transitions, or WSM actions. Its state upon power-up, after exit from deep power-down or after V_{CC} transitions below V_{LKO}, is read array mode.

After write or block erase is complete, the CUI must be reset to read array mode via the Read Array command, if access to the memory array is desired.

Storage Temperature ..... –30°C to +80°C

Voltage on Any Pin with
Respect to Ground..... –2.0V to V_{CC} +2.0V

V_{CC} Supply Voltage with
Respect to Ground..... –0.5V to +7.0V

beyond the "Operating Conditions" may effect device reliability.

NOTES:

1. Operating temperature is for commercial product defined by this specification.
2. Minimum DC input voltage is –0.5V. During transitions, inputs may undershoot to -2.0V for periods less than 20 ns. Maximum DC voltage on output pins is V_{CC} +0.5V, which may overshoot to V_{CC} + 2.0V for periods less than 20 ns.

Symbol	Parameter	Min	Max	Units
V_{CC}	V_{CC} Supply Voltage (± 0.25V)	4.75	5.25	V

9.0 VALUE SERIES 100 CARD DC CHARACTERISTICS

9.1 General

Symbol	Parameter	Notes	Min	Max	Units	Test Conditions
I_{LI}	Input Leakage Current	1,2		± 20	µA	$V_{CC} = V_{CC}$ Max $V_{IN} = V_{CC}$ or GND
I_{LO}	Output Leakage Current	1		± 20	µA	$V_{CC} = V_{CC}$ Max $V_{OUT} = V_{CC}$ or GND
V_{IL}	Input Low Voltage	1	0	0.8	V	
V_{IH}	Input High Voltage	1	2.4	V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage	1		0.4	V	I_{OL} = 3.2 mA
V_{OH}	Output High Voltage	1	V_{CC} – 0.4	V_{CC}	V	I_{OH} = –2.0 mA
V_{LKO}	V_{CC} Erase/Write Lock Voltage	1	2.0		V	

NOTES:

1. Values are the same for byte and word wide modes for all card densities.
2. Exceptions: With V_{IN} = GND, the leakage current on CE_1#, CE_2#, OE#, and WE# will be < 500 µA due to internal pull-up resistors. With $V_{IN} = V_{CC}$, RST leakage current will be < 150 µA due to internal pull-down resistors.

9.2 CMOS Interfacing: V_{CC} = 5.0V

DC Characteristics(1)

Sym	Parameter	Density (Mbytes)	Notes	x16 Mode Typ	x16 Mode Max	Units	Test Conditions
I_{CCR}	V_{CC} Read Current	2, 4, 8			75	mA	V_{CC} = V_{CC} Max t_{CYCLE} = 100 ns
I_{CCW}	V_{CC} Write Current	2, 4, 8			100	mA	
I_{CCE}	V_{CC} Erase Current	2, 4, 8			100	mA	
I_{CCSL}	V_{CC} Sleep Current	2, 4, 8	2	50	100	µA	V_{CC} = V_{CC} Max Control Signals = Vcc RESET = V_{IH}
I_{CCS}	V_{CC} Standby Current	2, 4, 8	2	3	10	mA	V_{CC} = V_{CC} Max Control Signals = V_{CC}

CMOS Test Conditions: V_{IL} = GND ± 0.2V, V_{IH} = V_{CC} ± 0.2V

NOTES:

1. All currents are RMS values unless otherwise specified. Typical V_{CC} = 5V, T = +25°C.
2. Control Signals: CE_1#, CE_2#, OE#, WE#.

9.3 TTL Interfacing: V_{CC} = 5.0V

DC Characteristics(1)

Sym	Parameter	Density (Mbytes)	Notes	x16 Mode Typ	x16 Mode Max	Units	Test Conditions
I_{CCR}	V_{CC} Read Current	2, 4, 8			100	mA	V_{CC} = V_{CC} Max t_{CYCLE} = 100 ns
I_{CCW}	V_{CC} Write Current	2, 4, 8	2		150	mA	
I_{CCE}	V_{CC} Erase Current	2, 4, 8	2		150	mA	
I_{CCSL}	V_{CC} Sleep Current	2, 4, 8	2		100	mA	V_{CC} = V_{CC} Max Control Signals = V_{IH}
I_{CCS}	V_{CC} Standby Current	2, 4, 8	2		100	mA	V_{CC} = V_{CC} Max Control Signals = V_{IH}

TTL Test Conditions: V_{IL}= 0.8V, V_{IH} = 2.4V

NOTES:

1. All currents are RMS values unless otherwise specified. Typical V_{CC} = 5V, T = +25°C.
2. Control Signals: CE_1#, CE_2#, OE#, WE#, RESET.

10.0 AC CHARACTERISTICS

AC timing diagrams and characteristics are guaranteed to meet or exceed PCMCIA 2.1 specifications. No delay occurs when switching between the common and attribute memory planes.

10.1 Read Operations: Common Memory

Symbol		Parameter	100 ns at 5V		Unit
JEDEC	PCMCIA		Min	Max	
t_{AVAV}	t_{RC}	Read Cycle Time	100		ns
t_{AVQV}	t_a (A)	Address Access Time		100	ns
t_{ELQV}	t_a (CE)	Card Enable Access Time		100	ns
t_{GLQV}	t_a (OE)	Output Enable Access Time		50	ns
t_{EHQX}	t_{dis} (CE)	Output Disable Time from CE#		50	ns
t_{GHQZ}	t_{dis} (OE)	Output Disable Time from OE#		50	ns
t_{ELQX}	t_{en} (CE)	Output Enable Time from CE#	5		ns
t_{GLQX}	t_{en} (OE)	Output Enable Time from OE#	5		ns
t_{PHQV}		Power-Down Recovery to Output Delay. V_{CC} = 5V		530	ns

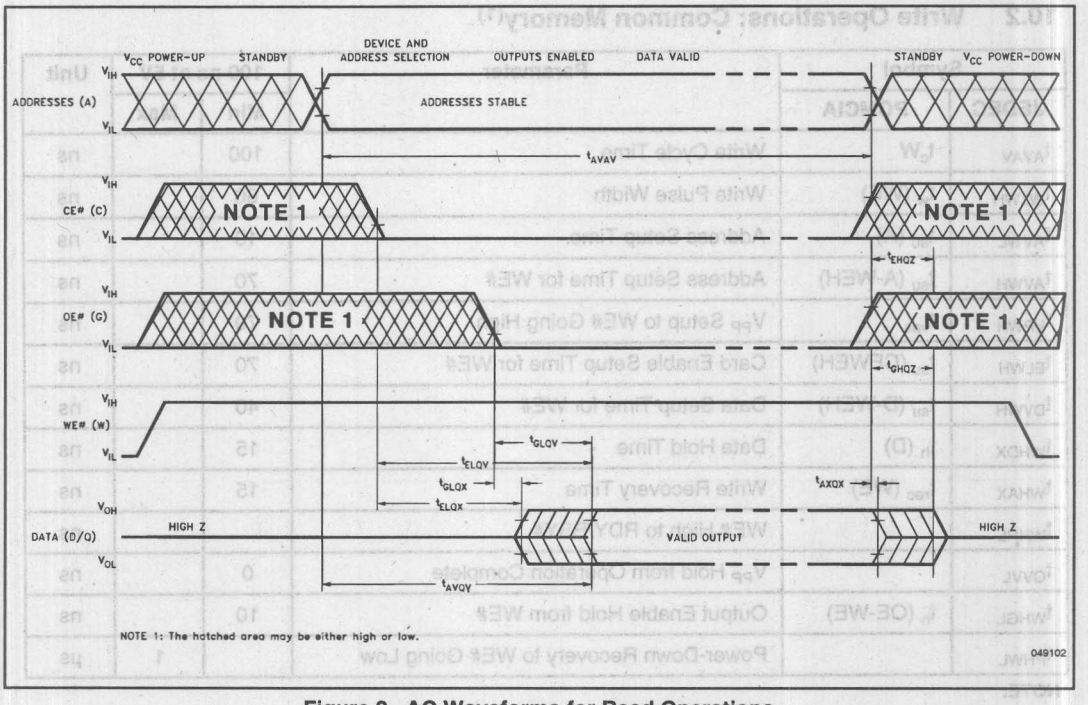


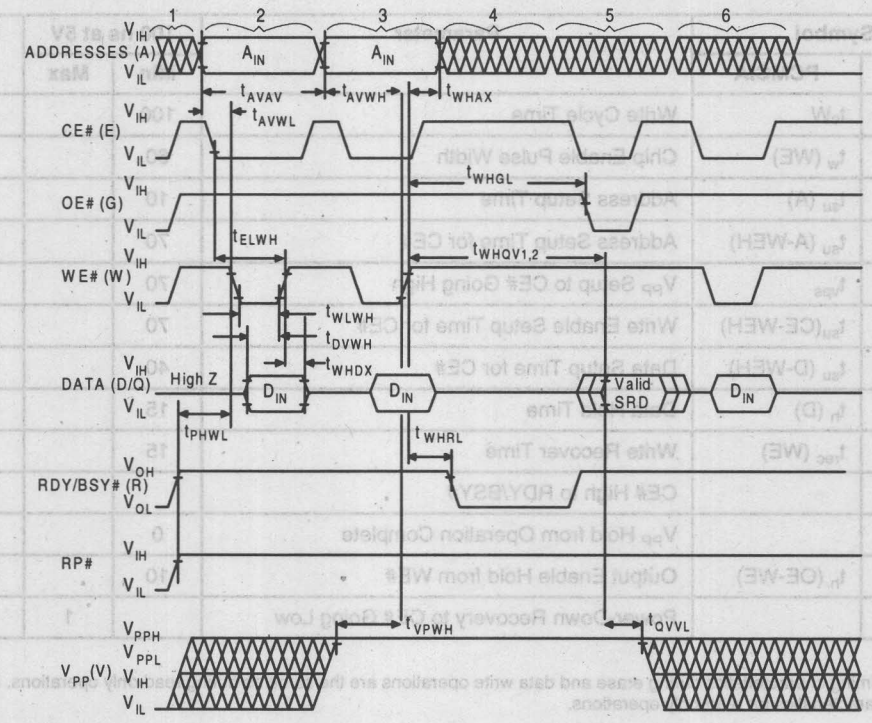
Figure 2. AC Waveforms for Read Operations

10.2 Write Operations: Common Memory(1)

Symbol		Parameter	100 ns at 5V		Unit
JEDEC	PCMCIA		Min	Max	
t_{AVAV}	t_cW	Write Cycle Time	100		ns
t_{WLWH}	t_w (WE)	Write Pulse Width	60		ns
t_{AVWL}	t_{su} (A)	Address Setup Time	10		ns
t_{AVWH}	t_{su} (A-WEH)	Address Setup Time for WE#	70		ns
t_{VPWH}	t_{vps}	V_{PP} Setup to WE# Going High	70		ns
t_{ELWH}	t_{su} (CEWEH)	Card Enable Setup Time for WE#	70		ns
t_{DVWH}	t_{su} (D-WEH)	Data Setup Time for WE#	40		ns
t_{WHDX}	t_h (D)	Data Hold Time	15		ns
t_{WHAX}	t_{rec} (WE)	Write Recovery Time	15		ns
t_{WHRL}		WE# High to RDY/BSY#			ns
t_{QVVL}		V_{PP} Hold from Operation Complete	0		ns
t_{WHGL}	t_h (OE-WE)	Output Enable Hold from WE#	10		ns
t_{PHWL}		Power-Down Recovery to WE# Going Low		1	µs

NOTE:

1. Read timing characteristics during erase and data write operations are the same as during read-only operations. Refer to AC Characteristics for read-only operations.



NOTES:

1. V_{CC} Power-Up and Standby
2. Write Data Write or Erase Setup Command
3. Write Valid Address and Data (Data Write) or Erase Confirm Command
4. Automated Data Write or Erase Delay
5. Read Status Register Data
6. Write Read Array Command

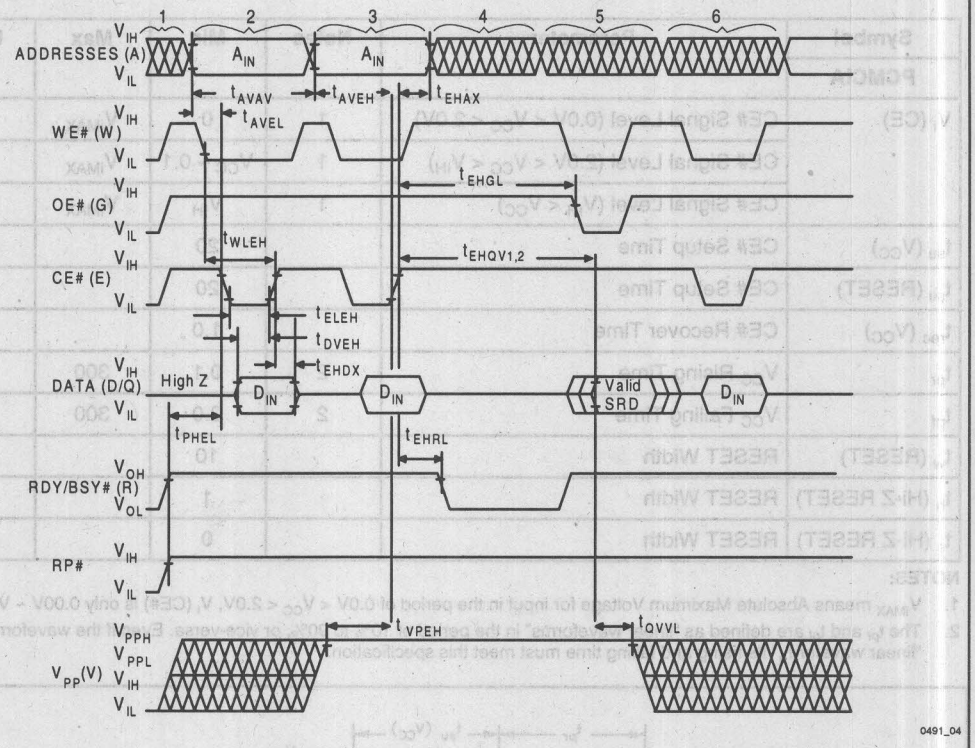
Figure 3. AC Waveforms for Write Operations

10.4 CE#-Controlled Write Operations: Common and Attribute Memory

Symbol		Parameter	100 ns at 5V		Unit
JEDEC	PCMCIA		Min	Max	
t_{AVAV}	t_cW	Write Cycle Time	100		ns
t_{ELEH}	t_w (WE)	Chip Enable Pulse Width	60		ns
t_{AVEL}	t_{su} (A)	Address Setup Time	10		ns
t_{AVEH}	t_{su} (A-WEH)	Address Setup Time for CE#	70		ns
t_{VPEH}	t_{vps}	V_{PP} Setup to CE# Going High	70		ns
t_{WLEH}	t_{su}(CE-WEH)	Write Enable Setup Time for CE#	70		ns
t_{DVEH}	t_{su} (D-WEH)	Data Setup Time for CE#	40		ns
t_{EHDX}	t_h (D)	Data Hold Time	15		ns
t_{EHAX}	t_{rec} (WE)	Write Recover Time	15		ns
t_{EHRL}		CE# High to RDY/BSY#			ns
t_{QVVL}		V_{PP} Hold from Operation Complete	0		ns
t_{EHGL}	t_h (OE-WE)	Output Enable Hold from WE#	10		ns
t_{PHEL}		Power-Down Recovery to CE# Going Low		1	µs

NOTE:

1. Read timing characteristics during erase and data write operations are the same as during read-only operations. Refer to AC Characteristics for read-only operations.



NOTES:

1. V_{CC} Power-Up and Standby
2. Write Data Write or Erase Setup Command
3. Write Valid Address and Data (Data Write) or Erase Confirm Command
4. Automated Data Write or Erase Delay
5. Read Status Register Data
6. Write Read Array Command

Figure 4. Alternate AC Waveform for Write Operations

10.5 Power-Up/Power-Down

Symbol	Parameter	Notes	Min	Max	Units
PCMCIA					
V_i (CE)	CE# Signal Level (0.0V < V_{CC} < 2.0V)	1	0	V_{iMAX}	V
	CE# Signal Level (2.0V < V_{CC} < V_{IH})	1	V_{CC} – 0.1	V_{iMAX}	V
	CE# Signal Level (V_{IH} < V_{CC})	1	V_{IH}	V_{iMAX}	V
t_{su} (V_{CC})	CE# Setup Time		20		ms
t_{su} (RESET)	CE# Setup Time		20		ms
t_{rec} (V_{CC})	CE# Recover Time		1.0		µs
t_{pr}	V_{CC} Rising Time	2	0.1	300	ms
t_{pf}	V_{CC} Falling Time	2	3.0	300	ms
t_w (RESET)	RESET Width		10		µs
t_h (Hi-Z RESET)	RESET Width		1		ms
t_s (Hi-Z RESET)	RESET Width		0		ms

NOTES:

1. V_{iMAX} means Absolute Maximum Voltage for input in the period of 0.0V < V_{CC} < 2.0V, V_i (CE#) is only 0.00V ~ V_{iMAX}.
2. The t_{pr} and t_{pf} are defined as "linear waveforms" in the period of 10% to 90%, or vice-versa. Even if the waveform is not a "linear waveform," its rising and falling time must meet this specification.

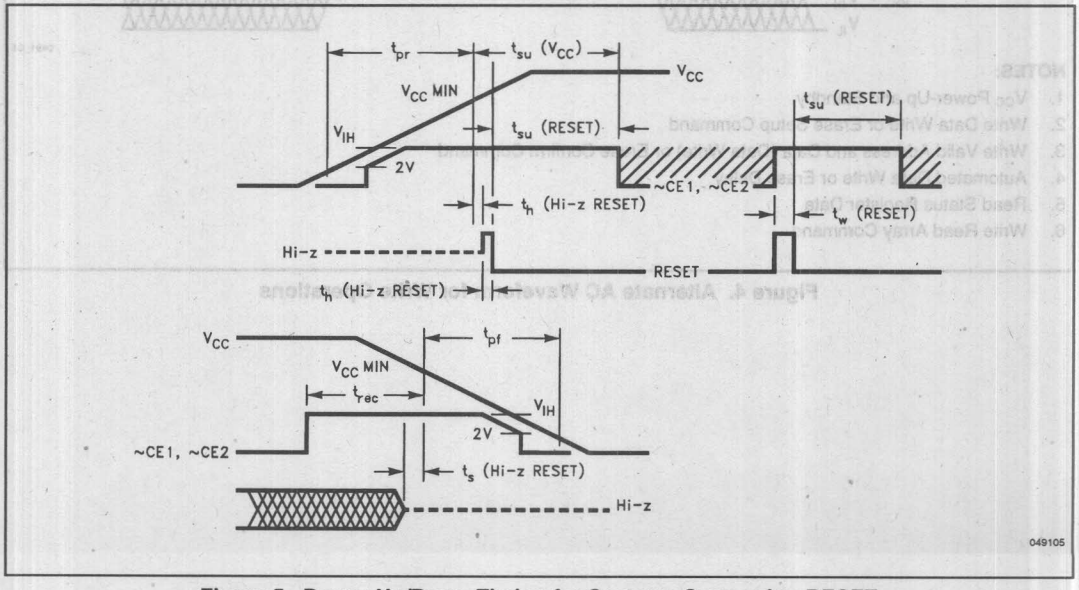


Figure 5. Power-Up/Down Timing for Systems Supporting RESET

11.0 CAPACITANCE

T_A = +25°C, f = 1.0 MHz

Symbol	Pins	Typ	Max	Unit
C_{IN}	Address/Control	25	50	pF
C_{IN}	V_{CC}	3	5	µF
C_{OUT}	Output	25	50	pF

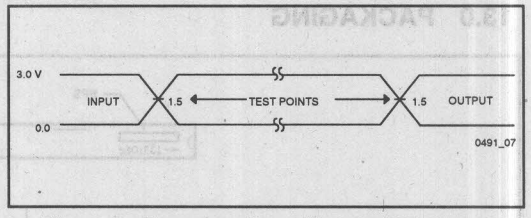


Figure 6. Transient Input/Output Reference Waveform for Standard Test Configuration

12.0 ERASE AND DATA WRITE PERFORMANCE(1,3)

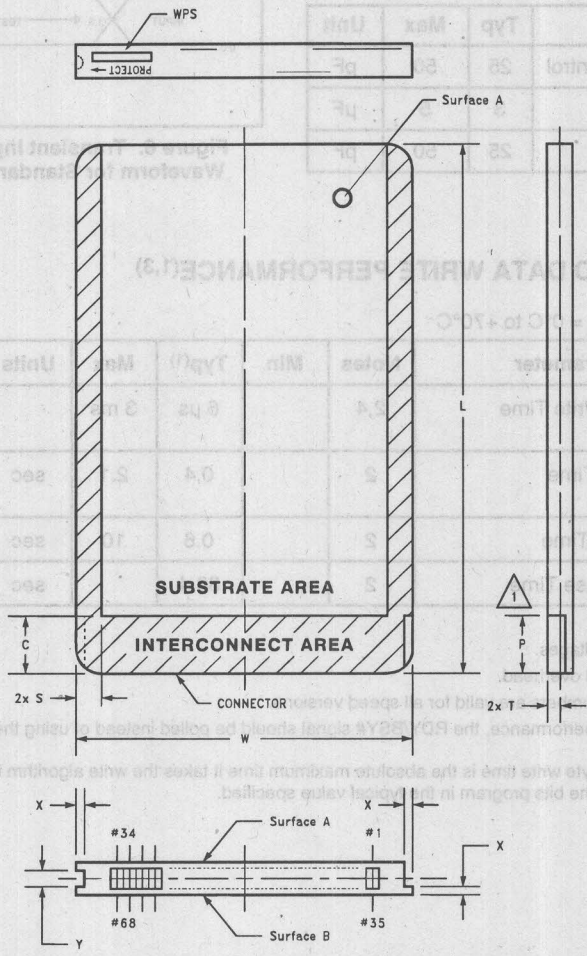
V_{CC} = 5.0V ± 0.5V, T_A = 0°C to +70°C

Sym	Parameter	Notes	Min	Typ(1)	Max	Units	Test Conditions
t_{WHQV1} t_{EHQV1}	Word/Byte Write Time	2,4		6 µs	3 ms		
t_{WHQV2} t_{EHQV2}	Block Write Time	2		0.4	2.1	sec	Word Write Mode
	Block Erase Time	2		0.6	10	sec	
	Full Chip Erase Time	2		38.4		sec	

NOTES:

1. +25°C, and normal voltages.
2. Excludes system-level overhead.
3. These performance numbers are valid for all speed versions.
4. To maximize system performance, the RDY/BSY# signal should be polled instead of using the maximum byte/word write time as a delay timer.
   The maximum word/byte write time is the absolute maximum time it takes the write algorithm to complete. The overwhelming majority of the bits program in the typical value specified.

13.0 PACKAGING



C MIN	L ± 0.008	P MIN ⚠1	S MIN	T ⚠2	W ± 0.004	X ± 0.002	Y ± 0.002
0.294 (10.0)	3.370 (85.60)	0.394 (10.0)	0.118 (3.0)	0.065 (1.65)	2.126 (54.0)	0.039 (1.00)	0.063 (1.60)

⚠1 POLARIZATION KEY LENGTH.

⚠2 INTERCONNECT AREA TOLERANCE = ±0.002
SUBSTRATE AREA TOLERANCE = ±0.004

3 MILLIMETERS ARE IN PARENTHESIS ().

049106

14.0 ADDITIONAL INFORMATION

14.1 Ordering Information

iMC008FLSC, SBXXXXX

Where:

i	= INTEL
MC	= MEMORY CARD
008	= DENSITY IN MEGABYTES (002, 004,008 AVAILABLE)
FL	= FLASH TECHNOLOGY
S	= BLOCKED ARCHITECTURE
C	= REVISION
SBXXXXX	= CUSTOMER IDENTIFIER

14.2 References

Order Number	Document
290429	28F008SA 8-Mbit (1 Mb x 8) FlashFile™ Memory Datasheet
292095	AP-360, "28F008SA Software Drivers"
292158	AP-606, "Interchangeablility of Series 1, Series 2, and Series 2+ Flash Memory Cards"
292177	AP-622, "Value Series 100 Card Design"
294016	ER-33, "ETOX™ IV Flash Memory Technology: Insight to Intel's Fourth Generation Process Innovation"

SERIES 2+ FLASH MEMORY CARDS
4-, 8-, 20- AND 40-MEGABYTE

iMC004FLSP, iMC008FLSP, iMC020FLSP, iMC040FLSP

- **Single Power Supply**
- **Automatically Reconfigures for 3.3V and 5V Systems**
- **150 ns Maximum Access Time with 5V Power Supply**
- **250 ns Maximum Access Time with 3.3V Power Supply**
- **High-Performance Random Writes**
  - **— 0.85 MB/S Sustained Throughput**
  - **— 1 KB Burst Write at 10 MB/S**
- **12 µA Typical Deep Power-Down**
- **Revolutionary Architecture**
  - **— Pipelined Command Execution**
  - **— Write during Erase**
  - **— Series 2 Command Super-Set**
- **State-of-the-Art 0.6 µm ETOX™ IV Flash Technology**
- **1 Million Erase Cycles per Block**
- **Up to 640 Independent Lockable Blocks**
- **PCMCIA 2.1/JEIDA 4.1-Compatible**
- **PCMCIA Type 1 Form Factor**
- **Series 2+ User's Manual**

Intel's Series 2+ Flash Memory Card sets the new record for high-performance disk emulation and eXecute-In-Place (XIP) applications in mobile PCs and dedicated equipment. Manufactured with Intel's 28F016SA 16-Mbit (DD28F032SA 32-Mbit) FlashFile™ memory, this card takes advantage of a revolutionary architecture that provides innovative capabilities, low-power operation and very high read/write performance.

The Series 2+ card provides today's highest density, highest performance nonvolatile read/write solution for solid-state storage applications. These applications are enhanced further with this product's symmetrically-blocked architecture, extended MTBF, low-power 3.3V operation, built-in V_{PP} generator, and multiple block locking methods. The Series 2+ card's dual read and write voltages allow interchange between 3.3V and 5.0V systems.

Order Number: 290491-006

intel

SERIES 2+ FLASH MEMORY CARDS
4-, 8-, 20- AND 40-MEGABYTE

MC004FLSP, MC008FLSP, MC020FLSP, MC040FLSP

- Single Power Supply
- Automatically Reconfigures for 3.3V and 5V Systems
- 150 ns Maximum Access Time with 5V Power Supply
- 250 ns Maximum Access Time with 3.3V Power Supply
- High-Performance Random Writes
  - 0.85 MB/S Sustained Throughput
  - 1 KB Burst Write at 10 MB/S
- 12 µA Typical Deep Power-Down
- Revolutionary Architecture
  - Pipelined Command Execution
  - Write during Erase
  - Series 2 Command Superset
- State-of-the-Art 0.6 µm ETOX™ IV Flash Technology
- 1 Million Erase Cycles per Block
- Up to 640 Independent Lockable Blocks
- PCMCIA 2.1/JEIDA 4.1-Compatible
- PCMCIA Type 1 Form Factor
- Series 2+ User's Manual

Intel's Series 2+ Flash Memory Card sets the new record for high-performance disk emulation and eXecute-In-Place (XIP) applications in mobile PCs and dedicated equipment. Manufactured with Intel's 28F016SA 16-Mbit (DD28F032SA 32-Mbit) FlashFile™ memory, this card takes advantage of a revolutionary architecture that provides innovative capabilities, low-power operation and very high read/write performance.

The Series 2+ card provides today's highest density, highest performance nonvolatile read/write solution for solid-state storage applications. These applications are enhanced further with Intel's product's symmetrically blocked architecture, extended MTBF, low-power 3.3V operation, built-in V_{PP} generator, and multiple block locking methods. The Series 2+ card's dual read and write voltages allow interchange between 3.3V and 5.0V systems.

1.0 SCOPE OF DOCUMENT

The documentation for Intel's Series 2+ Flash Memory Card includes this datasheet and the *Series 2+ Flash Memory Card User's Manual* (297373). The datasheet provides all AC and DC characteristics (including timing waveforms) and a convenient reference for the device command set and the card's integrated registers (including the 28F016SA's status registers). The *Series 2+ Memory Card User's Manual* provides a complete description of the methods for using the card. It also contains the full list of software algorithms and flowcharts and a section for upgrading Intel's Series 2 Flash Memory Cards designs.

2.0 PRODUCT OVERVIEW

The 4-, 8-, and 20- Megabyte Series 2+ Flash Memory Cards each contain a flash memory array that consists of two, four, and ten 28F016SA TSOP memory devices, respectively. Each 28F016SA contains 32 distinct, individually-erasable, 64-Kbyte blocks. Therefore, the 4-, 8-, and 20-megabyte cards contain 64, 128 and 320 independantly lockable blocks, respectively.

The 40-Megabyte Series 2+ Flash Memory Cards contain a flash memory array that consists of ten DD28F032SA TSOP memory devices. Each DD28F032SA contains two 28F016SA die in a single package, resulting in 64 distinct, individually-erasable, 64-Kbyte blocks. The 40-megabyte cards have 640 independently lockable blocks.

The Series 2+ Card offers additional product features to those of the Series 2 Card family (refer to the iMC0XXFLSA datasheets). Some of the more notable card-level enhancements include: interchangeable operation at 3.3V or 5.0V, block locking and internal V_{PP} generation.

The Series 2+ card incorporates V_{CC} detect circuitry, referred to as SmartPower, to sense the voltage level present at the card interface. The card's control logic automatically configures its circuitry and the 28F016SA/DD28F032SA memory array accordingly. The Card Information Structure (CIS) reports that the card is 3.3V or 5.0V compatible. The card also detects the presence of 12.0V on the V_{PP} pin and passes this supply to each memory device. When the 12.0V power supply is unavailable, the card can generate the required V_{PP} via its internal V_{PP} generation circuitry, whether V_{CC} is 3.3V or 5.0V.

At the device level, internal algorithm automation allows Write and Erase operations to be executed using a two-write command sequence in the same way as the 28F008SA FlashFile memory in the Series 2 Card. A super-set of commands and additional performance enhancements have been added to the basic 28F008SA command set:

- Page Buffer Write to Flash results in writes up to four times faster than Series 2 Cards.
- Command Queueing permits the devices to receive new commands during the execution of the current command set.
- Automatic data writes during erase allows the 28F016SA to perform Write operations to one block of memory while performing an Erase on another block.
- Software locking of memory blocks provides a means to selectively protect code or data within the card.
- Erase all unlocked blocks provides a quick and simple method to sequentially erase all the blocks within a 28F016SA memory device.

The Series 2+ Card has two ways to put the flash devices into a sleep mode for reduced power consumption:

1. Issue a command to individual devices, referred to as the software-controlled sleep mode. The device will retain status register data contents and finish any operation in progress using this approach.

2. Write to the card's PCMCIA-compatible Configuration and Status Register to activate a Reset Power-Down to all devices simultaneously.

The card achieves its PCMCIA-compatible word-wide access by pairing the 28F016SA/ DD28F032SA devices resulting in an accessible memory block size of 64 Kwords. The card's decoding logic (contained within the ASICs) allows the system to write or read one word at a time, or one byte at a time by referencing the high or low byte. Erasure can be performed on the entire block pair (high and low byte simultaneously) or on the high and low portions separately. Although the 28F016SA/DD28F032SA support byte or word-wide

data access, the byte interface was utilized within the card to allow the delivery of higher performance benefits, such as doubling the effective page buffer size and write performance.

The Series 2+ Card's ASICs also contain the Component Management Registers that provide five control functions: Ready-Busy Mode Selection, Software Write Protection, Card Status, Voltage Control, and Soft Reset.

The memory card interface supports the Personal Computer Memory Card Industry Association (PCMCIA 2.10) and Japanese Electronics Industry Development Association (JEIDA 4.1) 68-pin card format. The Series 2+ Flash Card meets all PCMCIA/JEIDA Type 1 mechanical specifications.

3.0 SERIES 2+ ARCHITECTURE OVERVIEW

The Series 2+ Card consists of three major functional elements—the flash memory array, card control and SmartPower circuitry. The card control logic handles the interface between the flash memory array and the host system's PCMCIA signals. SmartPower circuitry provides the card's integrated V_{PP} generator and a means for detecting the socket's voltage levels.

3.1 Card Pinout and Pin Description

The 68-pin PCMCIA format provides the system interface for the Series 2+ Flash Memory Card (see Tables 1 and 2). The detailed specifications for this interface is described in the PCMCIA 2.10 Standard Specification. The Series 2+ Flash Card product family conforms to the requirements of previous PCMCIA Versions Release 1.0, Release 2.0 and Release 2.01 of the PC Card Standard. Release 2.10 redefined pins 43 and 57 as VS_1 and VS_2 (previously REFRESH and RFU, respectively).

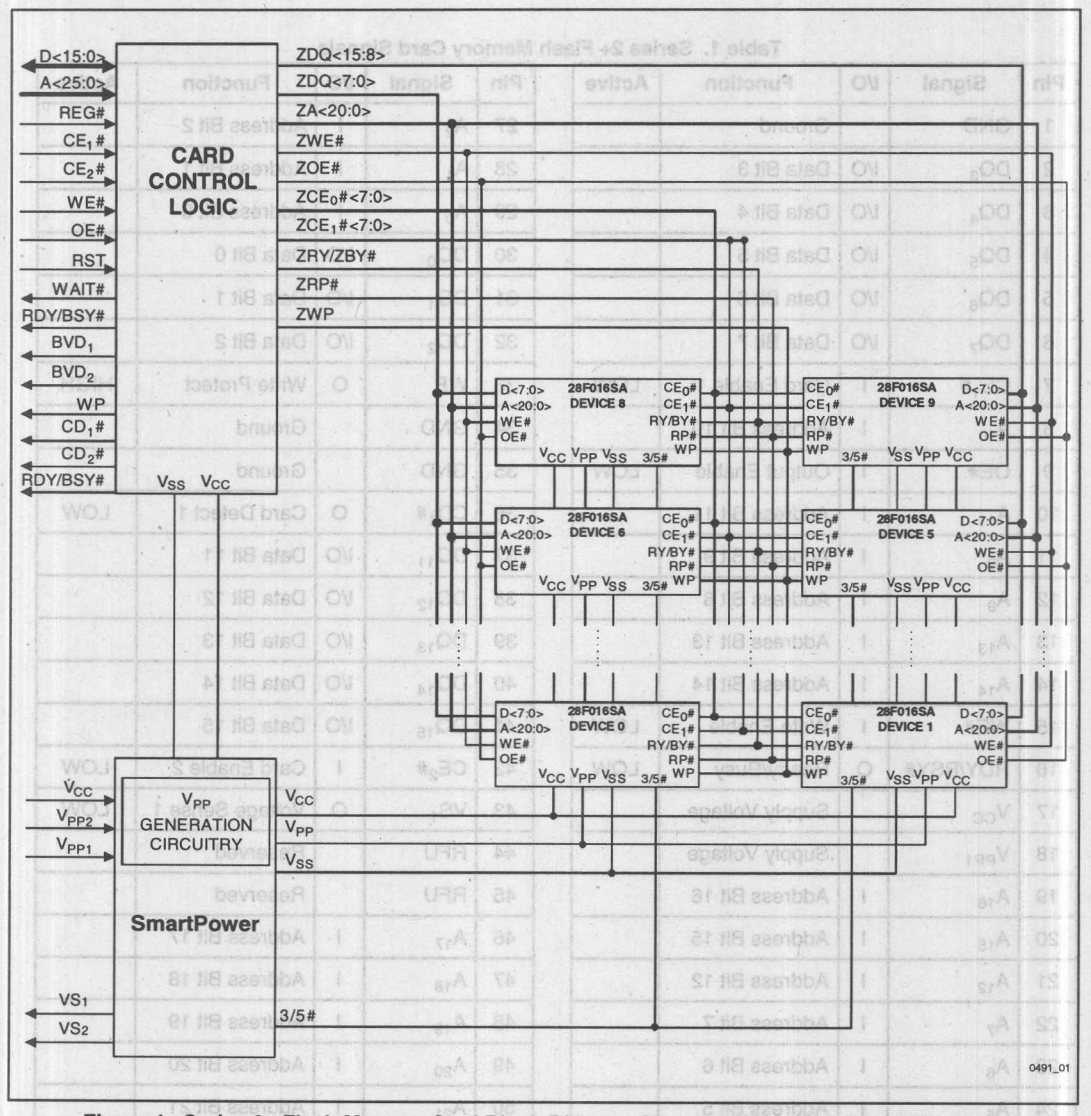


Figure 1. Series 2+ Flash Memory Card Block Diagram Showing Major Functional Elements

Table 1. Series 2+ Flash Memory Card Signals

Pin	Signal	I/O	Function	Active
1	GND		Ground	
2	DQ_3	I/O	Data Bit 3	
3	DQ_4	I/O	Data Bit 4	
4	DQ_5	I/O	Data Bit 5	
5	DQ_6	I/O	Data Bit 6	
6	DQ_7	I/O	Data Bit 7	
7	CE_1#	I	Card Enable 1	LOW
8	A_{10}	I	Address Bit 10	
9	OE#	I	Output Enable	LOW
10	A_{11}	I	Address Bit 11	
11	A_9	I	Address Bit 9	
12	A_8	I	Address Bit 8	
13	A_{13}	I	Address Bit 13	
14	A_{14}	I	Address Bit 14	
15	WE#	I	Write Enable	LOW
16	RDY/BSY#	O	Ready/Busy	LOW
17	V_{CC}		Supply Voltage	
18	V_{PP1}		Supply Voltage	
19	A_{16}	I	Address Bit 16	
20	A_{15}	I	Address Bit 15	
21	A_{12}	I	Address Bit 12	
22	A_7	I	Address Bit 7	
23	A_6	I	Address Bit 6	
24	A_5	I	Address Bit 5	
25	A_4	I	Address Bit 4	
26	A_3	I	Address Bit 3	

Pin	Signal	I/O	Function	Active
27	A_2	I	Address Bit 2	
28	A_1	I	Address Bit 1	
29	A_0	I	Address Bit 0	
30	DQ_0	I/O	Data Bit 0	
31	DQ_1	I/O	Data Bit 1	
32	DQ_2	I/O	Data Bit 2	
33	WP	O	Write Protect	HIGH
34	GND		Ground	
35	GND		Ground	
36	CD_1#	O	Card Detect 1	LOW
37	DQ_{11}	I/O	Data Bit 11	
38	DQ_{12}	I/O	Data Bit 12	
39	DQ_{13}	I/O	Data Bit 13	
40	DQ_{14}	I/O	Data Bit 14	
41	DQ_{15}	I/O	Data Bit 15	
42	CE_2#	I	Card Enable 2	LOW
43	VS_1	O	Voltage Sense 1	LOW
44	RFU		Reserved	
45	RFU		Reserved	
46	A_{17}	I	Address Bit 17	
47	A_{18}	I	Address Bit 18	
48	A_{19}	I	Address Bit 19	
49	A_{20}	I	Address Bit 20	
50	A_{21}	I	Address Bit 21	
51	V_{CC}		Supply Voltage	
52	V_{PP2}		Supply Voltage	

Table 1. Series 2+ Flash Memory Card Signals (Continued)

Pin	Signal	I/O	Function	Active
53	A_{22}	I	Address Bit 22	
54	A_{23}	I	Address Bit 23	
55	A_{24}	I	Address Bit 24	
56	A_{25}	I	Address Bit 25	
57	VS_2	O	Voltage Sense 2	N.C.
58	RST	I	Reset	HIGH
59	WAIT#	O	Extend Bus Cycle	LOW
60	RFU		Reserved	
61	REG#	I	Attribute Memory Select	LOW
62	BVD_2	O	Battery Voltage Detect 2	
63	BVD_1	O	Battery Voltage Detect 1	
64	DQ_8	I/O	Data Bit 8	
65	DQ_9	I/O	Data Bit 9	
66	DQ_{10}	I/O	Data Bit 10	
67	CD_2#	O	Card Detect 2	LOW
68	GND		Ground	

Table 2. Series 2+ Flash Memory Card Signal Description

Symbol	Type	Name and Function
A_0–A_{25}	INPUT	**ADDRESS INPUTS:** Address A_0 through A_{25} are address bus lines which enable direct addressing of up to 64 megabytes of memory on the card. Signal A_0 is not used in word access mode. A_{25} is the most significant bit.
DQ_0–DQ_{15}	INPUT/ OUTPUT	**DATA INPUT/OUTPUT:** DQ_0 through DQ_{15} constitute the bi-directional data bus. DQ_{15} is the most significant bit.
CE_1#,CE_2#	INPUT	**CARD ENABLE 1 & 2:** CE_1# enables even bytes, CE_2# enables odd bytes. Multiplexing A_0, CE_1# and CE_2# allows 8-bit hosts to access all data on D_0 through D_7.
OE#	INPUT	**OUTPUT ENABLE:** Active low signal gating read data from the memory card.
WE#	INPUT	**WRITE ENABLE:** Active low signal gating write data to the memory card.
RDY/BSY#	OUTPUT	**READY/BUSY OUTPUT:** Indicates status of internally timed erase or write activities. A high output indicates the memory card is ready to accept accesses. A low output indicates that a device in the memory card is busy with internally timed erase or write activities.
CD_1#,CD_2#	OUTPUT	**CARD DETECT 1 & 2:** These signals provide for correct memory card insertion detection. They are positioned at opposite ends of the card to detect proper alignment. The signals are connected to ground internally on the memory card, and will be forced low whenever a card is placed in the socket. The host socket interface circuitry shall supply 10K or larger pull-up resistors on these signal pins.

		These signals may be disconnected but are required for ExCA™ standard compliance.
V_{CC}		**CARD POWER SUPPLY:** (3.3V or 5V nominal) for all internal circuitry.
GND		**GROUND** for all internal circuitry.
REG#	INPUT	**REGISTER SELECT:** Provides access to Series 2+ Flash Memory Card registers and Card Information Structure in the Attribute Memory Plane.
RST	INPUT	**RESET:** Active high signal for placing card in Power-On Default State.
WAIT#	OUTPUT	**WAIT:** (Extend Bus Cycle) This signal is driven high for compatibility.
BVD_1, BVD_2	OUTPUT	**BATTERY VOLTAGE DETECT:** These signals are driven high to maintain SRAM card compatibility.
VS_1, VS_2	OUTPUT	**VOLTAGE SENSE:** Notify the host socket of the card's V_{CC} requirements. VS_1 grounded and VS_2 open indicates a 3.3V/5V card has been inserted.
RFU		**RESERVED FOR FUTURE USE**
N.C.		**NO INTERNAL CONNECTION TO CARD**; pin may be driven or left floating.

3.2 Series 2+ Card Control Logic

The Card Control Logic, contained within two ASICS, handles the address decoding and data control for the Series 2+ Card. The Component Management Registers are also contained within the Card Control Logic.

3.2.1 ADDRESS DECODE LOGIC

At the highest level, the Address Decode section determines when to select the Common Memory (REG# = V_{IH}) or Attribute Memory (REG# = V_{IL}) Planes. Within the Attribute Memory Plane (Figure 2) , the address decode logic determines when to select the Card Information Structure (CIS) or Component Management Registers (CMR). The CIS contains tuple information and is located at even-byte addresses beginning with address 0000H (refer to Section 6.0). The Component Management Registers are mapped at even byte locations beginning at address 4000H (refer to Section 3.3 for a detailed description).

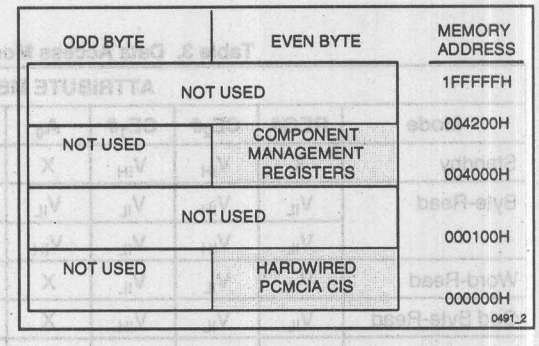


Figure 2. Attribute Memory Plane

3.2.2 DATA CONTROL

As shown in Table 3, data paths and directions are selected by the Data Control logic using REG#, A_0, WE#, OE#, CE_1#, and CE_2# as logic inputs. The Data Control logic selects any of the PCMCIA word-wide, byte-wide, and odd-byte modes for either Reads or Writes to Common or Attribute Memory. All accesses to the Attribute Memory Plane must be made through D[7:0] no valid data can be written on the high byte. Reads of D[15:8] will yield FFH.

Table 3. Data Access Mode Truth Table

COMMON MEMORY PLANE										
Mode	**REG#**	**CE_2#**	**CE_1#**	**A_0**	**OE#**	**WE#**	**V_{PP2}**	**V_{PP1}**	**D[15:8]**	**D[7:0]**
Standby	X	V_{IH}	V_{IH}	X	X	X	V_{PPL}	V_{PPL}	High-Z	High-Z
Byte-Read	V_{IH}	V_{IH}	V_{IL}	V_{IL}	V_{IL}	V_{IH}	V_{PPL}	V_{PPL}	High-Z	Even
	V_{IH}	V_{IH}	V_{IL}	V_{IH}	V_{IL}	V_{IH}	V_{PPL}	V_{PPL}	High-Z	Odd
Word-Read	V_{IH}	V_{IL}	V_{IL}	X	V_{IL}	V_{IH}	V_{PPL}	V_{PPL}	Odd	Even
Odd Byte-Read	V_{IH}	V_{IL}	V_{IH}	X	V_{IL}	V_{IH}	V_{PPL}	V_{PPL}	Odd	High-Z
Byte-Write	V_{IH}	V_{IH}	V_{IL}	V_{IL}	V_{IH}	V_{IL}	XXX	V_{PPH}	XXX	Even
	V_{IH}	V_{IH}	V_{IL}	V_{IH}	V_{IH}	V_{IL}	V_{PPH}	XXX	XXX	Odd
Word-Write	V_{IH}	V_{IL}	V_{IL}	X	V_{IH}	V_{IL}	V_{PPH}	V_{PPH}	Odd	Even
Odd Byte-Write	V_{IH}	V_{IL}	V_{IH}	X	V_{IH}	V_{IL}	V_{PPH}	V_{PPL}	Odd	XXX

Table 3. Data Access Mode Truth Table (Continued)

ATTRIBUTE MEMORY PLANE										
Mode	**REG#**	**CE_2#**	**CE_1#**	**A_0**	**OE#**	**WE#**	**V_{PP2}**	**V_{PP1}**	**$D_{[15:8]}$**	**$D_{[7:0]}$**
Standby	X	V_{IH}	V_{IH}	X	X	X	V_{PPL}	V_{PPL}	High-Z	High-Z
Byte-Read	V_{IL}	V_{IH}	V_{IL}	V_{IL}	V_{IL}	V_{IH}	V_{PPL}	V_{PPL}	High-Z	Even
	V_{IL}	V_{IH}	V_{IL}	V_{IH}	V_{IL}	V_{IH}	V_{PPL}	V_{PPL}	High-Z	FFH
Word-Read	V_{IL}	V_{IL}	V_{IL}	X	V_{IL}	V_{IH}	V_{PPL}	V_{PPL}	FFH	Even
Odd Byte-Read	V_{IL}	V_{IL}	V_{IH}	X	V_{IL}	V_{IH}	V_{PPL}	V_{PPL}	FFH	High-Z
Byte-Write	V_{IL}	V_{IH}	V_{IL}	V_{IL}	V_{IH}	V_{IL}	V_{PPL}	V_{PPL}	XXX	Even
	V_{IL}	V_{IH}	V_{IL}	V_{IH}	V_{IH}	V_{IL}	V_{PPL}	V_{PPL}	XXX	XXX
Word-Write	V_{IL}	V_{IL}	V_{IL}	X	V_{IH}	V_{IL}	V_{PPL}	V_{PPL}	XXX	Even
Odd Byte-Write	V_{IL}	V_{IL}	V_{IH}	X	V_{IH}	V_{IL}	V_{PPL}	V_{PPL}	XXX	XXX

NOTES:
When using the V_{PP} generator, V_{PP1} and V_{PP2} are "don't care."

3.3 Component Management Registers

The Component Management Registers (CMRs) are classified into two categories: those defined by PCMCIA Rev 2.0, and those included by Intel to enhance the interface between the host system and the card's flash memory array. The CMRs provide five control functions: Ready-Busy Mode selection, Voltage Control, Software Write Protection, Card Status and Soft Reset. For more details about the Component Management Register functionality, consult Intel's *Series 2+ Flash Memory Card User's Manual* (297373).

3.4 SmartPower

The SmartPower circuitry generates and monitors the card's programming voltages. When a host system does not provide a valid V_{PP} supply, the card's integrated generator can be switched on via the Voltage Control Register. The SmartPower circuitry also detects the host system's V_{CC} level (3.3V or 5.0V) and configures the card's flash memory devices, accordingly driving the 3/5# pin to the memory array to the appropriate value.

The SmartPower circuitry is enabled by writing a "1" to Bit 0 of the Voltage Control Register.

Table 4. Configuration Option Register - PCMCIA (Soft Reset Register)

Attribute Memory Plane Address: 4000H
Read/Write

SRESET	LevlREQ	Configuration Index					
7	6	5	4	3	2	1	0

Default: 02H

Bit 7 = Soft Reset
1 = Reset State
0 = End Reset Cycle

Bit 6 = Level Request
Driven Low

Bits 5-0 = Configuration Index
May Be Written with Values 1-4, Refer to Index in CIS Card Configuration Table Tuple.

Table 5. Card Configuration and Status Register - PCMCIA (Global Power-Down Register)

Attribute Memory Plane Address: 4002H
Read/Write

Reserved					PWRDWN	Reserved	
7	6	5	4	3	2	1	0

Default: 00H

Bit 2 = Power-Down
1 = 28F016SAs in Reset Power-Down
0 = Power-Up

Table 6. Card Status Register - Intel

Attribute Memory Plane Address: 4100H
Read Only

Reserved		SRESET	CMWP	PWRDWN	CISWP	WP	RDY/BSY#
7	6	5	4	3	2	1	0

Default: 01H or 03H

Bit 5 = Soft Reset
1 = Reset State

Bit 4 = Common Memory Write Protect
1 = Write Protected

Bit 3 = Power-Down
1 = Power-Down

Bit 2 = Common Memory CIS Write Protect
1 = Write Protected

Bit 1 = Write Protect Switch
1 = Write Protected

Bit 0 = Card Ready/Busy#
1 = Ready

Table 7. Write Protection Register - Intel

Attribute Memory Plane Address: 4104H
Read/Write

Reserved					BLKEN	CMWP	CISWP
7	6	5	4	3	2	1	0

Default: 04H

Bit 2 = Block Locking Enable
1 = Enable Independent 28F016SA Block Locking
0 = All Blocks Unlocked

Bit 1 = Common Memory Write Protect
1 = Common Memory Minus the CMCIS in Write Protect Status
0 = Write Protect According to Independent 28F016SA Block Locking

Bit 0 = Common Memory CIS Write Protect
1 = Common Memory CIS in Write Protect Status
0 = Write Protect According to Independent 28F016SA Block Locking

Table 8. Voltage Control Register - Intel

Attribute Memory Plane Address: 410CH
Read/Write

V_{CC} LEVEL	Reserved					V_{PP} VALID	V_{PP} GEN
7	6	5	4	3	2	1	0

Default: 82 or 02H

Bit 7 = V_{CC} Level: Read Only Bit
1 = Host Supplying 3.3V
0 = Host Supplying 5V

Bit 1 = V_{PP} Valid
1 = V_{PP} between 11.4V and 12V
0 = V_{PP} Invalid

Bit 0 = V_{PP} Generation
1 = Turn on Integrated V_{PP} Generator
0 = Turn off Integrated V_{PP} Generator

Note: The V_{PP} Valid bit only reflects the state of the V_{PP} Generator, NOT the external V_{PP}.

Table 9. Ready/Busy Mode Register - Intel

Attribute Memory Plane Address: 4140H
Read/Write

Reserved						RACK	MODE
7	6	5	4	3	2	1	0

Default: 00H

Bit 1 = Ready Acknowledge
0 = Clear RDY/BSY#

Bit 0 = RDY/BSY# Mode
1 = High-Performance Mode
0 = PCMCIA Level Mode

4.0 DEVICE COMMAND SET

The 28F016SA/DD28F032SA-based Series 2+ Command Set increases functionality over earlier 28F008SA-based designs while maintaining backwards compatibility. The extended command set incorporates many new features to improve programmability and write performance such as: page buffered writing, individual block locking, multiple RDY/BSY# configurations and device level queuing capabilities. The following pages list the Series 2+ command set and Bus Cycle Operations overview.

Series 2+ Command Set

Codes (H)	Series 2 Compatible Mode
00H	Invalid/Reserved
10H	Alternate Data Write
20H	Single Block Erase
40H	Data Write
50H	Clear Status Registers
70H	Read CSR
90H	Read ID Codes
B0H	Erase Suspend
D0H	Confirm/Resume
FFH	Read Flash Array

Codes (H)	Series 2+ Performance Enhancement
0CH	Page Buffer Write to Flash
71H	Read GSR or BSRs
72H	Page Buffer Swap
74H	Single Load to Page Buffer
75H	Read Page Buffer
77H	Lock Block
80H	Abort
96H,01H	RY/BY# Level Mode Enable
96H,02H	RY/BY# Pulse-On-Write
96H,03H	RY/BY# Pulse-On Erase
96H,04H	RY/BY# Disable
97H	Upload Status Bits
99H	Upload Device Information
A7H	Erase All Unlocked Blocks
E0H	Sequential Load to Page Buffer
F0H	Sleep

Table 10. 28F008SA-Compatible Mode Command Bus Definitions

Command	First Bus Cycle				Second Bus Cycle			
	R/W	Addr	Data		R/W	Addr	Data	
			Byte	Word			Byte	Word
Read Array	W	DA	FFH	FFFFH	R	DA	AD	AD
Intelligent Identifier	W	DA	90H	9090H	R	IA	ID	ID
Read CSR (See 1)	W	DA	70H	7070H	R	DA	CSRD	CSRD
Clear Status Register (See 2)	W	DA	50H	5050H				
Word/Byte Write §	W	WA	40H	4040H	W	WA	WD	WD
Word/Byte Write (Alternate) §	W	WA	10H	1010H	W	WA	WD	WD
Block Erase/Confirm §	W	BA	20H	2020H	W	BA	D0H	D0D0H
Erase Suspend/Resume	W	DA	B0H	B0B0H	W	DA	D0H	D0D0H

ADDRESSES:		DATA:	
DA	Device Address	AD	Array Data
BA	Block Address	CSRD	CSR Data
IA	Identifier Address	ID	Identifier Data
WA	Write Address	WD	Write Data

§ = Queueable Commands

NOTES:
1. The CSR is automatically available after the device enters Data Write, Erase or Suspend operations.
2. Clears CSR.3, CSR.4 and CSR.5. Also clears GSR.5 and all BSR.5 and BSR.2 bits.

Table 11. 28F016SA-Super-Set Mode Performance Enhancement Command Bus Definitions

Command	Notes	First Bus Cycle				Second Bus Cycle				Third Bus Cycle			
		Oper	Addr	Data		Oper	Addr	Data		Oper	Addr	Data	
				Byte	Word			Byte	Word			Byte	Word
Read Page Buffer		W	DA	75H	7575H	R	PA	PD	PDPD				
Page Buffer Swap	6	W	DA	72H	7272H								
Single Load to Page Buffer		W	DA	74H	7474H	W	PA	PD	PDPD				
Sequential Load to Page Buffer	4,5	W	DA	E0H	E0E0H	W	DA	BCH		W	DA	BCH	
Page Buffer Write to Flash Array §	3,4,5	W	DA	0CH	0C0CH	W	A_0	BC(L,H)		W	WA	BC(H,L)	
RY/BY# Pulse-On-Erase §	7	W	DA	96H	9696H	W	DA	D0H	D0D0H				
RY/BY# Pulse-On-Write §	7	W	DA	96H	9696H	W	DA	D1H	D1D1H				
RY/BY# Enable to Level-Mode §	7	W	DA	96H	9696H	W	DA	D2H	D2D2H				
RY/BY# Disable §	7	W	DA	96H	9696H	W	DA	D3H	D3D3H				
Lock Block/ Confirm §		W	DA	77H	7777H	W	BA	D0H	D0D0H				
Upload Status Bits/ Confirm §	2	W	DA	97H	9797H	W	DA	D0H	D0D0H				
Read Extended Status Registers	1	W	DA	71H	7171H	R	RA	GSRD/BSRD					
Erase All Unlocked Blocks/ Confirm §		W	DA	A7H	A7A7H	W	DA	D0H	D0D0H				
Sleep		W	DA	F0H	F0F0H								
Abort		W	DA	80H	8080H								
Upload Device Information		W	DA	99H	9999H	W	DA	D0H	D0D0H				

ADDRESSES		DATA		DATA COUNTS	
DA	Device Address	AD	Write Address	WC(L,H)	Word Count (Low, High)
BA	Block Address	CSRD	CSR Data	BC(L,H)	Byte Count (Low, High)
IA	Identifier Address	G/BSRD	GSR/BSR Data	WD(L,H)V	Write Data (Low, High)
PA	Page Buffer Address	ID	Identifier Data		
RA	Extended Register Address	WD	Write Data		
WA	Write Address	PD	Page Buffer Data		
X	Don't Care				

§ = Queueable Commands

NOTES:

1. RA can be the GSR address or any BSR address.
2. Upon device power-up, all BSR lock-bits are locked. The Lock Status Upload command must be written to reflect the actual lock-bit status.
3. A_0 is automatically complemented to load the second byte of data.
4. BCH/WCH must be at 00H for this product because of the 256-byte Page Buffer size *and* to avoid writing the Page Buffer contents into more than one 256-byte segment within an array block. They are simply shown for Page Buffer expandability.
5. PA and PD (whose count is given in cycles 2 and 3) are supplied starting in the fourth cycle (not shown).
6. This command allows the user to swap between available page buffers (0 or 1).
7. These commands reconfigure RY/BY# output to one of two pulse modes, or they enable and disable the RY/BY# function.

5.0 DEVICE STATUS REGISTER

Each 28F016SA has three types of status registers: the Compatible Status Register (CSR), the Global Status Register (GSR) and the Block Status Register (BSR). The Compatible Status Register is identical to the 28F008SA Status Register. The Global Status Register contains queue and page buffer information about each device. Each block within the device has a Block Status Register assigned to it. The Block Status Register contains the Block Locking Status and other information specific to the block being addressed.

Table 12. Compatible Status Register

Read Only Register

WSMS	ESS	ES	DWS	VPPS	Reserved		
7	6	5	4	3	2	1	0

Default: 80H

CSR.7 = WRITE STATE MACHINE STATUS (WSMS)
1 = Ready
0 = Busy

CSR.6 = ERASE-SUSPEND STATUS (ESS)
1 = Erase Suspended
0 = Erase In Progress/Completed

CSR.5 = ERASE STATUS (ES)
1 = Error In Block Erasure
0 = Successful Block Erase

CSR.4 = DATA-WRITE STATUS (DWS)
1 = Error in Data Write
0 = Data Write Successful

CSR.3 = V_{PP} STATUS (VPPS)
1 = V_{PP} Low Detect, Operation Abort
0 = V_{PP} OK

Table 13. Global Status Register

WSMS	ESS	ES	DWS	VPPS	Reserved		
7	6	5	4	3	2	1	0

Default: 80H

CSR.7 = WRITE STATE MACHINE STATUS (WSMS)
1 = Ready
0 = Busy

CSR.6 = ERASE-SUSPEND STATUS (ESS)
1 = Erase Suspended
0 = Erase In Progress/Completed

CSR.5 = ERASE STATUS (ES)
1 = Error In Block Erasure
0 = Successful Block Erase

CSR.4 = DATA-WRITE STATUS (DWS)
1 = Error in Data Write
0 = Data Write Successful

CSR.3 = V_{PP} STATUS (VPPS)
1 = V_{PP} Low Detect, Operation Abort
0 = V_{PP} OK

Table 14. Block Status Register

BS	BLS	BOS	BOAS	QS	VPPS	Reserved	
7	6	5	4	3	2	1	0

Default: 80H

BSR.7 = BLOCK STATUS (BS)
1 = Ready
0 = Busy

BSR.6 = BLOCK-LOCK STATUS (BLS)
1 = Block Unlocked for Write/Erase
0 = Block Locked for Write/Erase

BSR.5 = BLOCK OPERATION STATUS (BOS)
1 = Operation Unsuccessful
0 = Operation Successful or Currently Running

BSR.4 = BLOCK OPERATION ABORT STATUS (BOAS)
1 = Operation Aborted
0 = Operation Not Aborted

BSR.3 = QUEUE STATUS (QS)
1 = Queue Full
0 = Queue Available

BSR.2 = V_{PP} STATUS (VPPS)
1 = V_{PP} Low Detect, Operation Abort
0 = V_{PP} OK

6.0 PCMCIA CARD INFORMATION STRUCTURE

The Card Information Structure (CIS) begins at address 00000000H of the card's Attribute Memory Plane. It contains a variable length chain of data blocks (tuples) that conform to a basic format (Table 15). The CIS of the Series 2+ Flash Memory Card is found in Table 16.

Table 15. PCMCIA Tuple Format

Bytes	Data
0	Tuple Code: CISTPL_xxx. The tuple code 0FFH indicates no more tuples in the list.
1	Tuple Link: TPL_LINK. Link to the next tuple in the list. This can be viewed as the number of additional bytes in tuple, excluding this byte. A link field of zero indicates an empty tuple body. A link field containing 0FFH indicates the last tuple in the list.
2-n	Bytes specific to this tuple.

Table 16. Tuples for Series 2+ Card

Address	Value	Description
00h	01H	**CISTPL_DEVICE**
02h	04H	TPL_LINK
04h	57H	FLASH
06h	22H	150 ns
	32H	250 nx
08h		CARD SIZE
	0EH	4 MB
	1EH	8 MB
	4EH	20 MB
	9EH	40 MB
0AH	FFH	END OF DEVICE
0CH	1CH	**CISTPL_DEVICE_OC**
0EH	05H	TPL_LINK
10H	02H	OTHER CONDITIONS - 3 V_{CC}
12H	57H	FLASH
14H	32H	250 ns
16H		CARD SIZE
	0EH	4 MB
	1EH	8 MB
	4EH	20 MB
	9EH	40 MB
18H	FFH	END OF DEVICE
1AH	17H	**CISTPL_DEVICE_A**
1CH	04H	TPL_LINK
1EH	1FH	ROM
20H	22H	150 ns
22H	01H	2 Kb
24H	FFH	END OF DEVICE
26H	1DH	**CISTPL_DEVICE_OA**
28H	05H	TPL_LINK
2AH	02H	OTHER CONDITIONS - 3 V_{CC}
2CH	17H	ROM
2EH	32H	250 ns
30H	01H	2 Kb
32H	FFH	END OF DEVICE
34H	18H	**CISTPL JEDEC_C**
36H	02H	TPL_LINK
38H	89H	INTEL J-ID

Address	Value	Description
3AH	A0H	28F016 J-ID
3CH	00H	NULL CONTROL TUPLE
3EH	15H	**CISTPL VERS 1**
40H	39H	TPL_LINK
42H	04H	TPLLV1_MAJOR
44H	01H	TPLLV1_MINOR
		TPLLV1_INFO
46H	49H	I
48H	6EH	n
4AH	74H	t
4CH	65H	e
4EH	6CH	l
50H	00H	END TEXT
52H	53H	S
54H	32H	2
56H	45H	E
58H	34H	4 MB
	38H	8 MB
	32H	20 MB
	34H	40 MB
5AH	20H	4 MB
	20H	8 MB
	30H	20 MB
	30H	40 MB
5CH	53H	S
	20H	SPACE
5EH	57H	W
	20H	SPACE
60H	00H	ENDTEXT
62H	43H	C
64H	4FH	O
66H	50H	P
68H	59H	Y
6AH	52H	R
6CH	49H	I
6EH	47H	G
70H	48H	H
72H	54H	T

Table 16. Tuples for Series 2+ Card (Contd.)

Address	Value	Description
74H	20H	SPACE
76H	49H	I
78H	6EH	n
7AH	74H	t
7CH	65H	e
7EH	6CH	l
80H	20H	SPACE
82H	43H	C
84H	4FH	O
86H	52H	R
88H	50H	P
8AH	4FH	O
8CH	52H	R
8EH	41H	A
90H	54H	T
92H	49H	I
94H	4FH	O
96H	4EH	N
98H	20H	SPACE
9AH	31H	1
9CH	39H	9
9EH	39H	9
A0H	33H	3
A2H	20H	SPACE
A4H	47H	G
A6H	4CH	L
A8H	41H	A
AAH	44H	D
ACH	45H	E
AEH	4BH	K
B0H	00H	END TEXT
B2H	FFH	END OF LIST
B4H	1AH	**CISTPL_CONF**
B6H	05H	TUPL_LINK
B8H	01H	TPCC_SZ
BAH	04H	TPCC_LAST
BCH	00H	TPCC_RADR

Address	Value	Description
BEH	40H	TPCC_RADR
C0H	03H	TPCC_RMSK
C2H	00H	NULL CONTROL TUPLE
C4H	1BH	**CISTPL_CFTABLE_ENTRY**
C6H	08H	TPL_LINK
C8H	01H	**TPCE_INDEX (01H)**
CAH	01H	TPCE_FS (V_{CC} ONLY)
CCH	79H	TPCE_PD V_{CC} PARAMETER SELECTION BYTE
CEH	55H	V_{CC} NOMINAL VOLTAGE 5V ± 5%
D0H	53H	I_{CC} STATIC 500 µA
D2H	1EH	I_{CC} AVERAGE 150 mA
D4H	1EH	I_{CC} PEAK 150 mA
D6H	1BH	I_{CC} PWRDWN 200 µA
D8H	1BH	**CISTPL_CFTABLE_ENTRY**
DAH	0FH	TPL_LINK
DCH	02H	**TPCE_INDEX (02H)**
DEH	02H	TPCE_FS (V_{CC} AND V_{PP})
E0H	79H	TPCE_PD V_{CC} PARAMETER SELECTION BYTE
E2H	55H	V_{CC} NOMINAL VOLTAGE 5V ± 5%
E4H	2BH	I_{CC} STATIC 250 µA
E6H	06H	I_{CC} AVERAGE 100 mA
E8H	06H	I_{CC} PEAK 100 mA
EAH	52H	I_{CC} PWRDWN 50 µA
ECH	79H	TPCE_PD V_{PP} PARAMETER SELECTION BYTE
EEH	8EH	12.0V ± 5%
F0H	7DH	NC OK ON STANDBY & PWD
F2H	53H	I_{PP} STATIC 500 µA
F4H	25H	I_{PP} AVERAGE 20 mA
F6H	25H	I_{PP} PEAK 20 mA

Table 16. Tuples for Series 2+ Card (Contd.)

Address	Value	Description
F8H	52H	IPP PWRDWN 50 µA
FAH	1BH	**CISTPL_CFTABLE_ENTRY**
FCH	09H	TPL_LINK
FEH	03H	**TPCE_INDEX (03H)**
100H	01H	TPCE_FS (V_{CC} ONLY)
102H	79H	TPCE_PD V_{CC} PARAMETER SELECTION BYTE
104H	B5H	V_{CC} = 3.3V
106H	1EH	EXTENSION BYTE
108H	04H	I_{CC} STATIC 1 mA
10AH	1EH	I_{CC} AVERAGE 150 mA
10CH	1EH	I_{CC} PEAK 150 mA
10EH	53H	I_{CC} PWRDWN 500 µA
110H	1BH	**CISTPL_CFTABLE_ENTRY**
112H	10H	TPL_LINK
114H	04H	**TPCE_INDEX (04H)**
116H	02H	TPCE_FS (VCC AND VPP)
118H	79H	TPCE_PD V_{CC} PARAMETER SELECTION BYTE
11AH	B5H	V_{CC} = 3.3V
11CH	1EH	EXTENSION BYTE
11EH	2BH	I_{CC} STATIC 250 µA
120H	06H	I_{CC} AVERAGE 100 mA
122H	06H	I_{CC} PEAK 100 mA
124H	52H	I_{CC} PWRDWN 50 µA
126H	79H	TPCE_PD V_{PP} PARAMETER SELECTION BYTE
128H	8EH	12.0V =/- 5%
12AH	7DH	NC OK ON STANDBY & PWD
12CH	53H	I_{PP} STATIC 500 µA
12EH	25H	I_{PP} AVERAGE 20 mA
130H	25H	I_{PP} PEAK 20 mA
132H	1BH	I_{PP} PWRDWN 150 µA
134H	00H	NULL CONTROL TUPLE

Address	Value	Description
136H	00H	NULL CONTROL TUPLE
138H	1EH	**CISTPL DEVICEGEO**
13AH	06H	TPL_LINK
13CH	02H	DGTPL_BUS
13EH	11H	DGTPL_EBS
140H	01H	DGTPL_RBS
142H	01H	DGTPL_WBS
144H	01H	DGTPL_PART = 1
146H	01H	FLASH DEVICE INTERLEAVE
148H	20H	**CISTPL_MANFID**
14AH	04H	TPL_LINK (04H)
		TPLMID_MANF
14CH	89H	LSB
14EH	00H	MSB
150H	12H	4 MB - 150 ns
	22H	8 MB - 150 ns
	42H	20 MB - 150 ns
	62H	40 MB - 150 ns
152H	84H	TPLMID_CARD MSB
154H	21H	**CISTPL_FUNCID**
156H	02H	TPL_LINK
158H	01H	TPLFID_FUNCTION (MEMORY)
15AH	00H	TPLFID_SYSINIT (NONE)
15CH	FFH	**CISTPL_END**
	00H	INVALID ADDRESS
		(156H-1FEH)

7.0 SYSTEM DESIGN CONSIDERATIONS

7.1 Power Supply Decoupling

Flash memory power-switching characteristics require careful device decoupling. System designers are interested in three supply current issues: standby, active and transient current peaks which are produced by rising and falling edges of CE_1# and CE_2#. The capacitive and inductive loads on the card and internal flash memory device pairs determine the magnitudes of these peaks.

Three-line control and proper decoupling capacitor selection suppress transient voltage peaks. Series 2+ cards contain on-card ceramic decoupling capacitors connected between V_{CC} and GND, and between V_{PP1}/V_{PP2} and GND.

The card connector should also have a 4.7 µF electrolytic capacitor between V_{CC} and GND, as well as between V_{PP1}/V_{PP2} and GND. The bulk capacitors overcome voltage slumps caused by printed-circuit-board trace inductance, and supply charge to the smaller capacitors as needed.

7.2 Power-Up/Down Protection

The PCMCIA/JEIDA-specified socket properly sequences the power supplies to the flash memory card via shorter and longer pins. This design assures that hot insertion and removal will not result in card damage or data loss.

Each device in the memory card is designed to offer protection against accidental erasure or writing, caused by spurious system-level signals that may exist during power transitions. The card will power-up into the read state.

A system designer must guard against active writes for V_{CC} voltages above V_{LKO} when V_{PP} is active. Since both WE# and CE_1# must be low for a command write, driving either to V_{IH} will inhibit writes. With its control register architecture, alteration of device contents only occurs after successful completion of the two-step command sequences. While these precautions are sufficient for most applications, an alternative approach would allow V_{CC} to reach its steady state value before raising V_{PP1}/V_{PP2} above V_{CC} + 2.0V. In addition, upon powering down, V_{PP1}/V_{PP2} should be below V_{CC} + 2.0V before lowering V_{CC}.

NOTE

The Integrated V_{PP} generator defaults to the power off condition after reset and system power-up. The Vpp Generation circuitry must be enabled for the memory card to operate in 3.3V-Only or 5.0V-Only mode.

7.3 Hot Insertion/Removal

The capability to remove or insert PC cards while the system is powered on (i.e., hot insertion/removal) requires careful design approaches on the system and card levels. To design for this capability, consider card over-voltage stress, system power fluxuations and control line stability.

8.0 ELECTRICAL SPECIFICATIONS

NOTICE: This is a production datasheet. The specifications are subject to change without notice.

8.1 Absolute Maximum Ratings*

Operating Temperature
- During Read ..... 0°C to +60°C(1)
- During Write ..... 0°C to +60°C

Storage Temperature ..... -30°C to +70°C(2)

Voltage on Any Pin with Respect to Ground ..... -2.0V to V_{CC} +2.0V(2)

V_{PP1}/V_{PP2} Supply Voltage with Respect to Ground ..... -2.0V to V_{CC} +14.0V(2,3)

V_{CC} Supply Voltage with Respect to Ground ..... -0.5V to +7.0V

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

NOTES:
1. Operating temperature is for commercial product defined by this specification.
2. Minimum DC input voltage is -0.5V. During transitions, inputs may undershoot to 2.0V for periods less than 20 ns. Maximum DC voltage on output pins is V_{CC} +0.5V, which may overshoot to V_{CC} + 2.0V for periods less than 20 ns.
3. Maximum DC input voltage on V_{PP1}/V_{PP2} may overshoot to +14.0V for periods less than 20 ns.
4. V_{PP} generator turned "on" for 3.3V or 5.0V only operation.

Symbol	Parameter	Min	Max	Units
V_{CC} at 3.3V, 12V V_{pp}	V_{CC} Supply Voltage (±0.3V)	3.0	3.6	V
V_{CC} at 3.3V, V_{PP} Gen4	V_{CC} Supply Voltage (±0.15V)	3.15	3.45	V
V_{CC} at 5.0V	V_{CC} Supply Voltage (±0.25V)	4.75	5.25	V

9.0 SERIES 2+ DC CHARACTERISTICS

9.1 General

Symbol	Parameter	Notes	Min	Max	Units	Test Conditions
I_{LI}	Input Leakage Current	1,3		±20	µA	$V_{CC} = V_{CC}$ Max $V_{IN} = V_{CC}$ or GND
I_{LO}	Output Leakage Current	1		±20	µA	$V_{CC} = V_{CC}$ Max $V_{OUT}=V_{CC}$ or GND
V_{IL5}	Input Low Voltage	1	0	0.8	V	V_{CC} = 5V
$V_{IL3.3}$				0.7		V_{CC} = 3.3V
V_{IH5}	Input High Voltage	1	2.4	V_{CC} + 0.5	V	V_{CC} = 5V
$V_{IH3.3}$			2.2	V_{CC} + 0.3		V_{CC} = 3.3V
V_{OL}	Output Low Voltage	1		0.4	V	I_{OL} = 3.2 mA
V_{OH}	Output High Voltage	1	V_{CC} – 0.4	V_{CC}	V	I_{OH} = -2.0 mA
V_{PPL}	V_{PP} during Read Only Operations	1,2	0	6.5	V	
V_{PPH}	V_{PP} during Read/Write Operations	1	11.4	12.6	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage	1	2.0		V	

NOTES:

1. Values are the same for byte and word wide modes for all card densities.
2. Block Erases/Data Writes are inhibited when V_{PP} and V_{PPL} are not guaranteed in the range between V_{PPH} and V_{PPL}.
3. Exceptions: With V_{IN} = GND, the leakage current on CE_1#, CE_2#, REG#, OE#, and WE# will be < 500 µA due to internal pull-up resistors. With $V_{IN} = V_{CC}$, RST leakage current will be < 500 µA due to internal pull-down resistors. With $V_{IN} = V_{CC}$, A_{21}-A_{25} leakage current will be <100 µA due to internal pull down resistors.

9.2 CMOS Interfacing: V_{CC} = 3.3V

Series 2+ DC Characteristics (1)

Sym	Parameter	Density (Mbytes)	Notes	x8 Mode Typ	x8 Mode Max	x16 Mode Typ	x16 Mode Max	Unit	Test Conditions
I_{CCR}	V_{CC} Read Current	4, 8, 20, 40	2, 3		75		100	mA	$V_{CC} = V_{CC}$ Max t_{CYCLE} = 250 ns
I_{CCW}	V_{CC} Write Current	4, 8, 20, 40	2, 3, 4		40		50	mA	V_{PP} Gen = OFF during Data Write
			2, 3, 5		100		175	mA	V_{PP} Gen = ON during Data Write
I_{CCE}	V_{CC} Erase Current	4, 8, 20, 40	2, 3, 4		40		50	mA	V_{PP} Gen = OFF
			2, 3, 5		80		150	mA	V_{PP} Gen = ON
I_{CCSL}	V_{CC} Sleep Current	4	3, 4, 6	12	20	12	20	µA	$V_{CC} = V_{CC}$ Max
		8		14	30	14	30		Control Signals =
		20		20	60	20	60		V_{CC}
		40		30	110	30	110		
I_{CCS}	V_{CC} Standby Current	4	2, 3, 4, 6	75	115	110	210	µA	$V_{CC} = V_{CC}$ Max
		8		80	125	115	230		Control Signals =
		20		85	155	120	250		V_{CC}
		40		100	200	150	300		
I_{PPW}	V_{PP} Write Current	4, 8, 20, 40	2, 3, 4	10	15	20	30	mA	Data Write in Progress $V_{PP} = V_{PPH}$
I_{PPE}	V_{PP} Erase Current	4, 8, 20, 40	2, 3, 4	6	12	12	22	mA	Block (Pair) Erase in Progress $V_{PP} = V_{PPH}$
I_{PPSL}	V_{PP} Sleep Current	4, 8, 20, 40	2, 3, 4	0	0.5	0	0.5	mA	$V_{PP} \leq V_{CC}$
I_{PPS}	V_{PP} Standby/ Read Current	4, 8, 20, 40	2, 3, 4	0	0.5	0	0.5	mA	$V_{PP} \leq V_{CC}$

CMOS Test Conditions: V_{IL} = GND ± 0.2V $V_{IH} = V_{CC}$ ± 0.2V

NOTES:

1. All currents are RMS values unless otherwise specified. Typical V_{CC} = 5V, V_{PP} = 12V, T= 25°C.
2. Two devices active in word mode, one device active in byte mode.
3. Devices not addressed are in sleep mode.
4. V_{PP} Generation Circuitry turned off.
5. V_{PP} Generation Circuitry turned on.
6. Control Signals, CE_1#, CE_2#, OE#, WE#, REG#.

9.3 CMOS Interfacing: V_{CC} = 5.0V

Series 2+ DC Characteristics (1)

Sym	Parameter	Density (Mbytes)	Notes	x8 Mode Typ	x8 Mode Max	x16 Mode Typ	x16 Mode Max	Unit	Test Conditions
I_{CCR}	V_{CC} Read Current	4, 8, 20, 40	2, 3		140		160	mA	$V_{CC} = V_{CC}$ Max t_{CYCLE} = 250 ns
I_{CCW}	V_{CC} Write Current	4, 8, 20, 40	2, 3, 4		85		120	mA	V_{PP} Gen = OFF during Data Write
			2, 3, 5		120		150	mA	V_{PP} Gen = ON during Data Write
I_{CCE}	V_{CC} Erase Current	4, 8, 20, 40	2, 3, 4		75		100	mA	V_{PP} Gen = OFF
			2, 3, 5		100		150	mA	V_{PP} Gen = ON
I_{CCSL}	V_{CC} Sleep Current	4	3, 4, 6	12	20	12	20	µA	$V_{CC} = V_{CC}$ Max
		8		14	30	14	30		Control Signals =
		20		20	60	20	60		V_{CC}
		40		30	110	30	110		
I_{CCS}	V_{CC} Standby Current	4	2, 3, 4, 6	75	115	110	210	µA	$V_{CC} = V_{CC}$ Max
		8		80	125	115	230		Control Signals =
		20		85	155	120	250		V_{CC}
		40		100	200	150	300		
I_{PPW}	V_{PP} Write Current	4, 8, 20, 40	2, 3, 4	6	12	14	24	mA	Data Write in Progress $V_{PP} = V_{PPH}$
I_{PPE}	V_{PP} Erase Current	4, 8, 20, 40	2, 3, 4	6	12	12	22	mA	Block (Pair) Erase in Progress $V_{PP} = V_{PPH}$
I_{PPSL}	V_{PP} Sleep Current	4, 8, 20, 40	2, 3, 4	0	0.5	0	0.5	mA	$V_{PP} \leq V_{CC}$
I_{PPS}	V_{PP} Standby/ Read Current	4, 8, 20, 40	2, 3, 4	0	0.5	0	0.5	mA	$V_{PP} \leq V_{CC}$

CMOS Test Conditions: V_{IL} = GND ± 0.2V, $V_{IH} = V_{CC}$ ± 0.2V

NOTES:

1. All currents are RMS values unless otherwise specified. Typical V_{CC} = 5V, V_{PP} = 12V, T = 25°C.
2. Two devices active in word mode, one device active in byte mode.
3. Devices not addressed are in sleep mode.
4. V_{PP} Generation Circuitry turned off.
5. V_{PP} Generation Circuitry turned on.
6. Control Signals, CE_1#, CE_2#, OE#, WE#, REG#.

9.4 TTL Interfacing: V_{CC} = 3.3V

Series 2+ DC Characteristics(1)

Sym	Parameter	Density	Notes	x8 Mode		x16 Mode		Unit	Test Conditions
		(Mbytes)		Typ	Max	Typ	Max		
I_{CCR}	V_{CC} Read Current	4, 8, 20, 40	2, 3		75		90	mA	V_{CC} = V_{CC} Max t_{CYCLE} = 250 ns
I_{CCW}	V_{CC} Write Current	4, 8, 20, 40	2, 3, 4		85		100	mA	V_{PP} Gen = OFF during Data Write
			2, 3, 5		150		225	mA	V_{PP} Gen = ON during Data Write
I_{CCE}	V_{CC} Erase Current	4, 8, 20, 40	2, 3, 4		85		100	mA	V_{PP} Gen = OFF
			2, 3, 5		125		180	mA	V_{PP} Gen = ON
I_{CCSL}	V_{CC} Sleep Current	4, 8, 20, 40	3, 4, 6		70		70	mA	V_{CC} = V_{CC} Max Control Signals = V_{IH}
I_{CCS}	V_{CC} Standby Current	4, 8, 20, 40	2, 3, 4, 6		70		70	mA	V_{CC} = V_{CC} Max Control Signals = V_{IH}
I_{PPW}	V_{PP} Write Current	4, 8, 20, 40	2, 3, 4	10	15	20	30	mA	Data Write in Progress V_{PP} = V_{PPH}
I_{PPE}	V_{PP} Erase Current	4, 8, 20, 40	2, 3, 4	5	10	10	20	mA	Block (Pair) Erase in Progress V_{PP} = V_{PPH}
I_{PPSL}	V_{PP} Sleep Current	4, 8, 20, 40	2, 3, 4	0	0.5	0	0.5	mA	$V_{PP} \leq V_{CC}$
I_{PPS}	V_{PP} Standby/ Read Current	4, 8, 20, 40	2, 3, 4	0	0.5	0	0.5	mA	$V_{PP} \leq V_{CC}$

TTL Test Conditions: V_{IL} = 0.7V, V_{IH} = 2.2V

NOTES:

1. All currents are RMS values unless otherwise specified. Typical V_{CC} = 5V, V_{PP} = 12V, T= 25°C.
2. Two devices active in word mode, one device active in byte mode.
3. Devices not addressed are in sleep mode.
4. V_{PP} Generation Circuitry turned off.
5. V_{PP} Generation Circuitry turned on.
6. Control Signals, CE_1#, CE_2#, OE#, WE#, REG#.

9.5 TTL Interfacing: V_{CC} = 5.0V

Series 2+ DC Characteristics(1)

Sym	Parameter	Density (Mbytes)	Notes	x8 Mode		x16 Mode		Unit	Test Conditions
				Typ	Max	Typ	Max		
I_{CCR}	V_{CC} Read Current	4, 8, 20, 40	2, 3		170		190	mA	$V_{CC} = V_{CC}$ Max t_{CYCLE} = 150 ns
I_{CCW}	V_{CC} Write Current	4, 8, 20, 40	2, 3, 4		135		170	mA	V_{PP} Gen = OFF during Data Write
			2, 3, 5		170		250	mA	V_{PP} Gen = ON during Data Write
I_{CCE}	V_{CC} Erase Current	4, 8, 20, 40	2, 3, 4		125		150	mA	V_{PP} Gen = OFF
			2, 3, 5		150		200	mA	V_{PP} Gen = ON
I_{CCSL}	V_{CC} Sleep Current	4, 8, 20, 40	3, 4, 6		100		100	mA	$V_{CC} = V_{CC}$ Max Control Signals = V_{IH}
I_{CCS}	V_{CC} Standby Current	4, 8, 20, 40	2, 3, 4, 6		100		100	mA	$V_{CC} = V_{CC}$ Max Control Signals = V_{IH}
I_{PPW}	V_{PP} Write Current	4, 8, 20, 40	2, 3, 4	7	12	14	24	mA	Data Write in Progress $V_{PP} = V_{PPH}$
I_{PPE}	V_{PP} Erase Current	4, 8, 20, 40	2, 3, 4	5	10	10	20	mA	Block (Pair) Erase in Progress $V_{PP} = V_{PPH}$
I_{PPSL}	V_{PP} Sleep Current	4, 8, 20, 40	2, 3, 4	0	0.5	0	0.5	mA	$V_{PP} \leq V_{CC}$
I_{PPS}	V_{PP} Standby/ Read Current	4, 8, 20, 40	2, 3, 4	0	0.5	0	0.5	mA	$V_{PP} \leq V_{CC}$

TTL Test Conditions: V_{IL} = 0.8V, V_{IH} = 2.4V

NOTES:

1. All currents are RMS values unless otherwise specified. Typical V_{CC} = 5V, V_{PP} = 12V, T= 25°C.
2. Two devices active in word mode, one device active in byte mode.
3. Devices not addressed are in sleep mode.
4. V_{PP} Generation Circuitry turned off.
5. V_{PP} Generation Circuitry turned on.
6. Control Signals, CE_1#, CE_2#, OE#, WE#, REG#.

10.0 AC CHARACTERISTICS

AC Timing Diagrams and characteristics are guaranteed to meet or exceed PCMCIA 2.1 specifications. No delay occurs when switching between the Common and Attribute Memory Planes.

10.1 Read Operations: Common and Attribute Memory

Symbol		Parameter	150 ns at 5V		250 ns at 3.3V		Unit
JEDEC	PCMCIA		Min	Max	Min	Max	
t_{AVAV}	t_{RC}	Read Cycle Time	150		250		ns
t_{AVQV}	t_a (A)	Address Access Time		150		250	ns
t_{ELQV}	t_a (CE)	Card Enable Access Time		150		250	ns
t_{GLQV}	t_a (OE)	Output Enable Access Time		75		125	ns
t_{EHQX}	t_{dis} (CE)	Output Disable Time from CE#		75		100	ns
t_{GHQZ}	t_{dis} (OE)	Output Disable Time from OE#		75		100	ns
t_{GLQX}	t_{en} (CE)	Output Enable Time from CE#	5		5		ns
t_{ELQX}	t_{en} (OE)	Output Enable Time from OE#	5		5		ns
t_{PHQV}		Power-Down Recovery to Output Delay. V_{CC} = 5V		530		670	ns

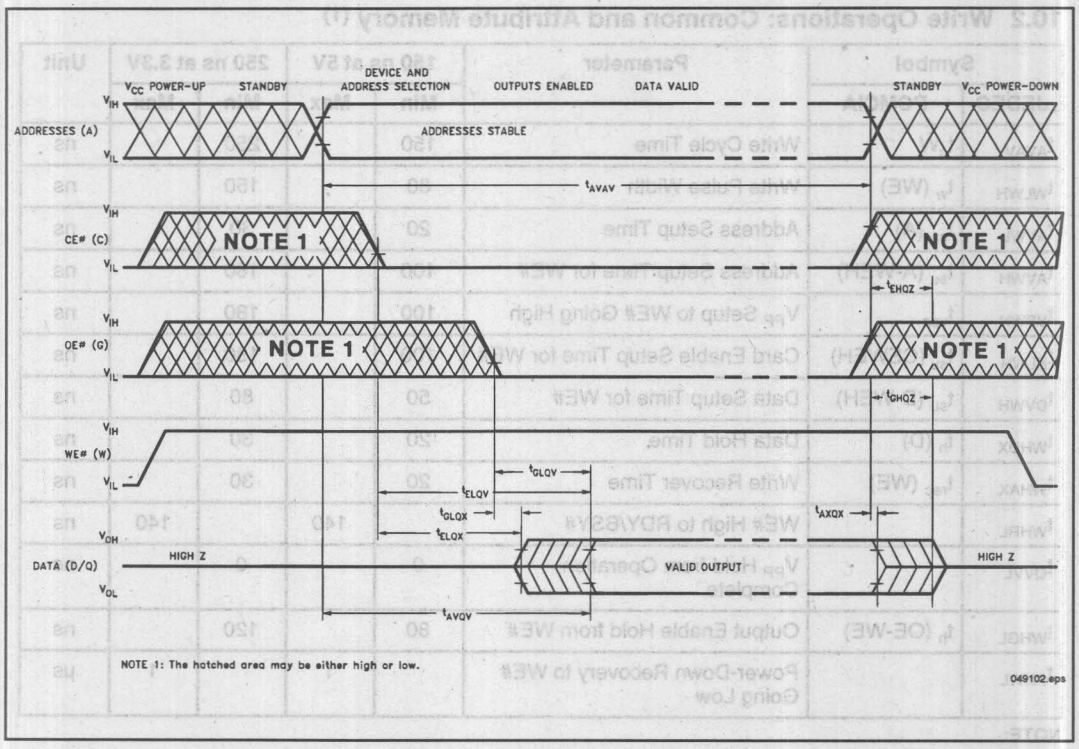


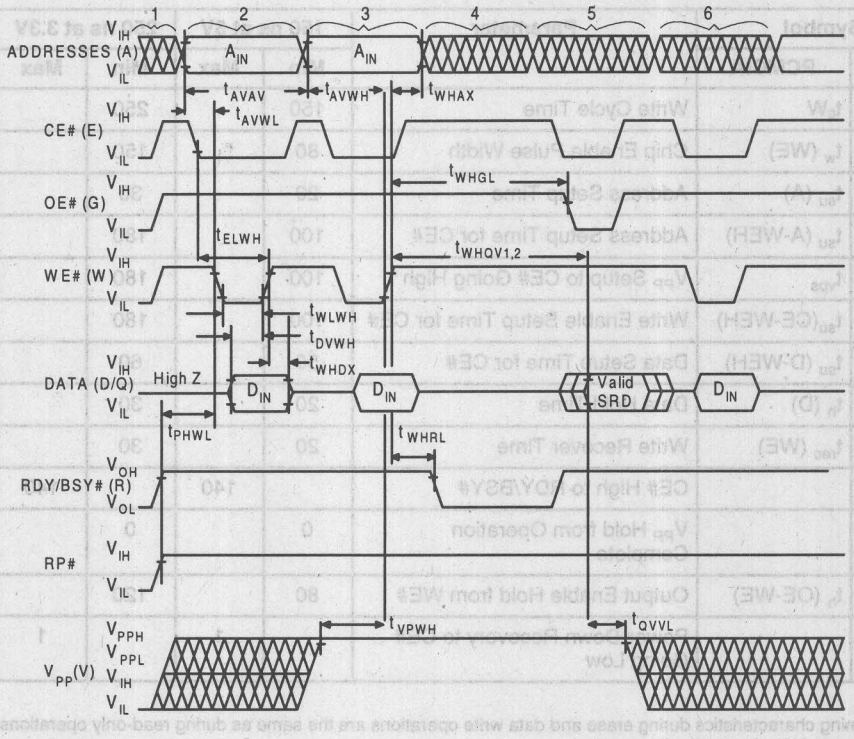
Figure 3. AC Waveforms for Read Operations

10.2 Write Operations: Common and Attribute Memory (1)

Symbol		Parameter	150 ns at 5V		250 ns at 3.3V		Unit
JEDEC	PCMCIA		Min	Max	Min	Max	
t_{AVAV}	t_cW	Write Cycle Time	150		250		ns
t_{WLWH}	t_w (WE)	Write Pulse Width	80		150		ns
t_{AVWL}	t_{su} (A)	Address Setup Time	20		30		ns
t_{AVWH}	t_{su} (A-WEH)	Address Setup Time for WE#	100		180		ns
t_{VPWH}	t_{vps}	V_{PP} Setup to WE# Going High	100		180		ns
t_{ELWH}	t_{su} (CEWEH)	Card Enable Setup Time for WE#	100		180		ns
t_{DVWH}	t_{su} (D-WEH)	Data Setup Time for WE#	50		80		ns
t_{WHDX}	t_h (D)	Data Hold Time	20		30		ns
t_{WHAX}	t_{rec} (WE)	Write Recover Time	20		30		ns
t_{WHRL}		WE# High to RDY/BSY#		140		140	ns
t_{QVVL}		V_{PP} Hold from Operation Complete	0		0		ns
t_{WHGL}	t_h (OE-WE)	Output Enable Hold from WE#	80		120		ns
t_{PHWL}		Power-Down Recovery to WE# Going Low		1		1	µs

NOTE:

1. Read timing characteristics during erase and data write operations are the same as during read-only operations. Refer to AC Characteristics for read-only operations.



NOTES:

1. V_{CC} Power-Up and Standby
2. Write Data Write or Erase Setup Command
3. Write Valid Address and Data (Data Write) or Erase Confirm Command
4. Automated Data Write or Erase Delay
5. Read Status Register Data
6. Write Read Array Command

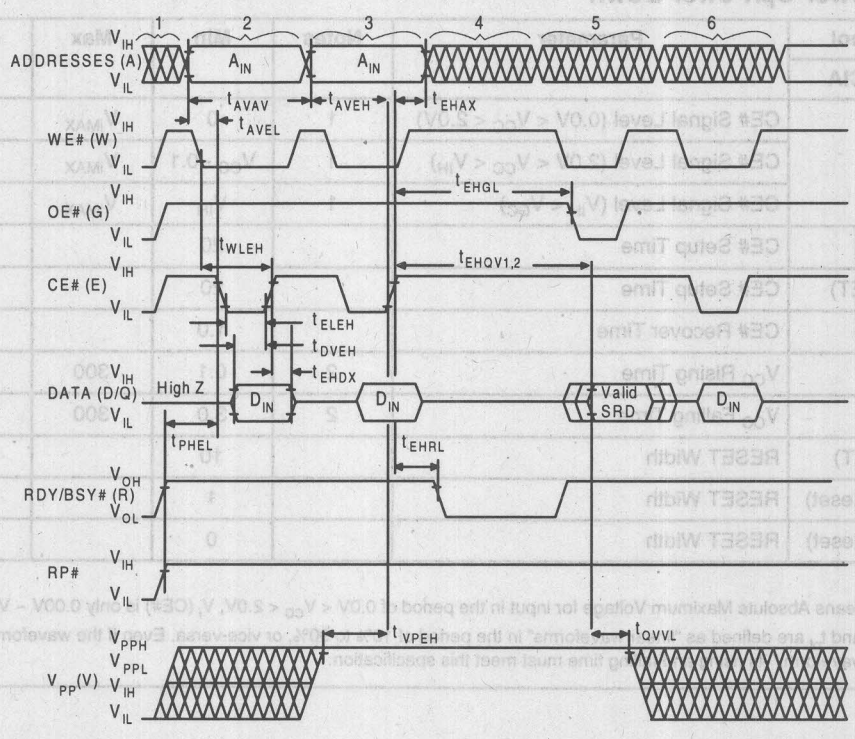
Figure 4. AC Waveforms for Write Operations

10.3 CE#-Controlled Write Operations: Common and Attribute Memory

Symbol		Parameter	150 ns at 5V		250 ns at 3.3V		Unit
JEDEC	PCMCIA		Min	Max	Min	Max	
t_{AVAV}	t_cW	Write Cycle Time	150		250		ns
t_{ELEH}	t_w (WE)	Chip Enable Pulse Width	80		150		ns
t_{AVEL}	t_{su} (A)	Address Setup Time	20		30		ns
t_{AVEH}	t_{su} (A-WEH)	Address Setup Time for CE#	100		180		ns
t_{VPEH}	t_{vps}	V_{PP} Setup to CE# Going High	100		180		ns
t_{WLEH}	t_{su}(CE-WEH)	Write Enable Setup Time for CE#	100		180		ns
t_{DVEH}	t_{su} (D-WEH)	Data Setup Time for CE#	50		60		ns
t_{EHDX}	t_h (D)	Data Hold Time	20		30		ns
t_{EHAX}	t_{rec} (WE)	Write Recover Time	20		30		ns
t_{EHRL}		CE# High to RDY/BSY#		140		140	ns
t_{QVVL}		V_{PP} Hold from Operation Complete	0		0		ns
t_{EHGL}	t_h (OE-WE)	Output Enable Hold from WE#	80		120		ns
t_{PHEL}		Power-Down Recovery to CE# Going Low		1		1	µs

NOTE:

1. Read timing characteristics during erase and data write operations are the same as during read-only operations. Refer to AC Characteristics for read-only operations.



NOTES:

1. V_{CC} Power-Up and Standby
2. Write Data Write or Erase Setup Command
3. Write Valid Address and Data (Data Write) or Erase Confirm Command
4. Automated Data Write or Erase Delay
5. Read Status Register Data
6. Write Read Array Command

Figure 5. Alternate AC Waveform for Write Operations

10.4 Power-Up/Power-Down

Symbol	Parameter	Notes	Min	Max	Units
PCMCIA					
V_i (CE)	CE# Signal Level (0.0V < V_{CC} < 2.0V)	1	0	V_{iMAX}	V
	CE# Signal Level (2.0V < V_{CC} < V_{IH})	1	V_{CC} - 0.1	V_{iMAX}	V
	CE# Signal Level (V_{IH} < V_{CC})	1	V_{IH}	V_{iMAX}	V
t_{su} (V_{CC})	CE# Setup Time		20		ms
t_{su} (RESET)	CE# Setup Time		20		ms
t_{rec} (V_{CC})	CE# Recover Time		1.0		µs
t_{pr}	V_{CC} Rising Time	2	0.1	300	ms
t_{pf}	V_{CC} Falling Time	2	3.0	300	ms
t_w (RESET)	RESET Width		10		µs
t_h (Hi-Z Reset)	RESET Width		1		ms
t_s (Hi-Z Reset)	RESET Width		0		ms

NOTES:

1. V_{iMAX} means Absolute Maximum Voltage for input in the period of 0.0V < V_{CC} < 2.0V, V_i (CE#) is only 0.00V ~ V_{iMAX}.
2. The t_{pr} and t_{pf} are defined as "linear waveforms" in the period of 10% to 90%, or vice-versa. Even if the waveform is not a "linear waveform," its rising and falling time must meet this specification.

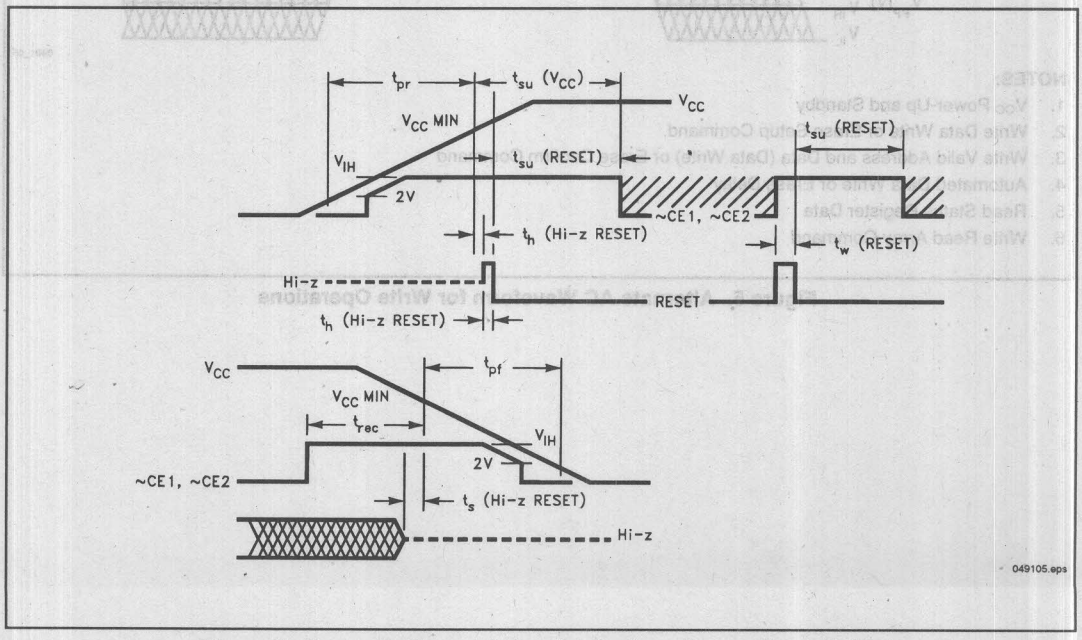


Figure 6. Power-Up/Down Timing for Systems Supporting RESET

10.5 Series 2+ Card Information Structure Tuple Changes

Address	Value	Description
06H	22H	150 ns
	32H	250 ns
14H	32H	250 ns
150H	12H	4 MB - 150 ns
	22H	8 MB - 150 ns
	42H	20 MB - 200 ns

11.0 CAPACITANCE

T_A = 25°C, f = 1.0 MHz

Symbol	Pins	Typ	Max	Unit
C_{IN}	A_0	15	30	pF
C_{IN}	Address/Control	10	20	pF
C_{IN}	V_{CC}, V_{PP}	2	2	µF
C_{OUT}	Output	10	20	pF

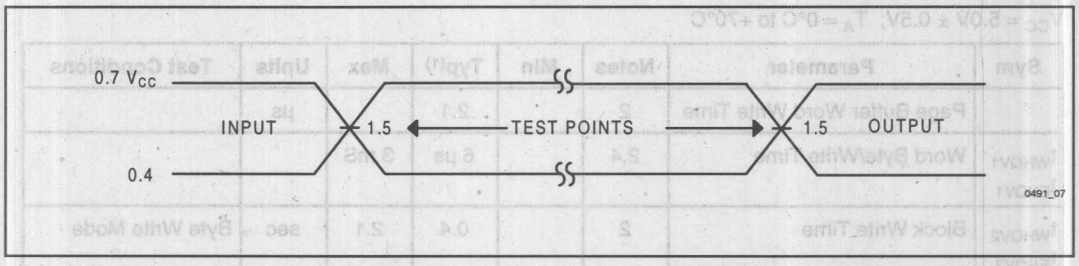


Figure 7. Transient Input/Output Reference Waveform (V_{CC} = 5.0V) for Standard Test Configuration

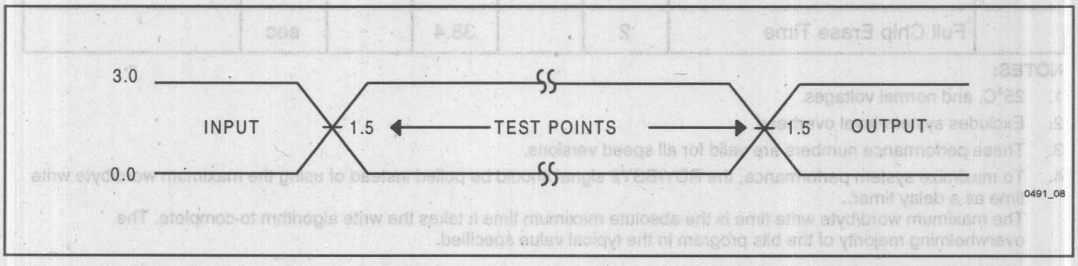


Figure 8. Transient Input/Output Reference Waveform (V_{CC} = 3.3V) for Standard Test Configuration

12.0 ERASE AND DATA WRITE PERFORMANCE (1,3)

V_{CC} = 3.3V ± 0.3V, T_A = 0°C to +70°C

Sym	Parameter	Notes	Min	Typ(1)	Max	Units	Test Conditions
	Page Buffer Word Write Time	2		2.2		µs	
t_{WHQV1} t_{EHQV1}	Word/Byte Write Time	2		9 µs	3 mS		
t_{WHQV2} t_{EHQV2}	Block Write Time	2		0.6	2.1	sec	Byte Write Mode
	Block Erase Time	2		0.8	10	sec	
	Full Chip Erase Time	2		51.2		sec	

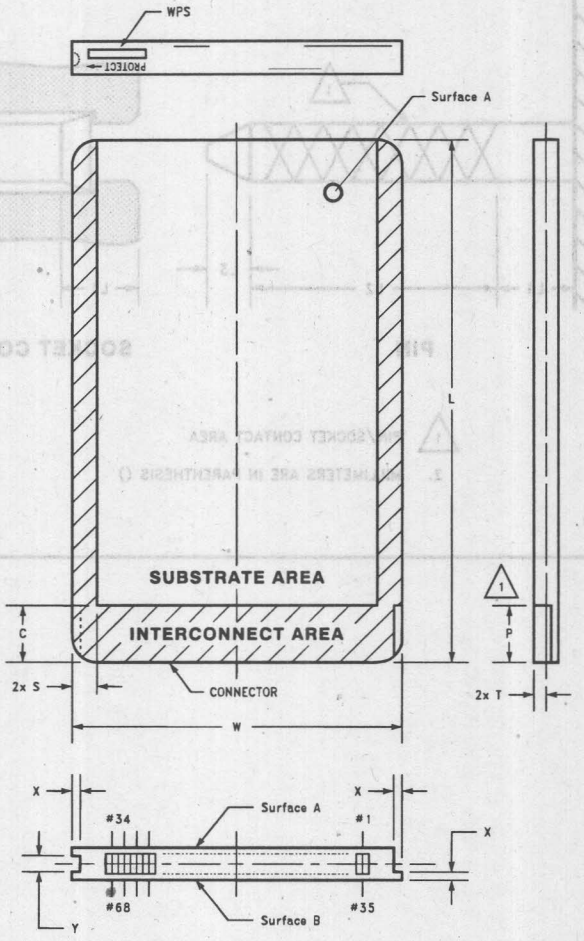
V_{CC} = 5.0V ± 0.5V, T_A = 0°C to +70°C

Sym	Parameter	Notes	Min	Typ(1)	Max	Units	Test Conditions
	Page Buffer Word Write Time	2		2.1		µs	
t_{WHQV1} t_{EHQV1}	Word Byte/Write Time	2,4		6 µs	3 mS		
t_{WHQV2} t_{EHQV2}	Block Write Time	2		0.4	2.1	sec	Byte Write Mode
	Block Erase Time	2		0.6	10	sec	
	Full Chip Erase Time	2		38.4		sec	

NOTES:

1. 25°C, and normal voltages.
2. Excludes system-level overhead.
3. These performance numbers are valid for all speed versions.
4. To maximize system performance, the RDY/BSY# signal should be polled instead of using the maximum word/byte write time as a delay timer.
   The maximum word/byte write time is the absolute maximum time it takes the write algorithm to complete. The overwhelming majority of the bits program in the typical value specified.

13.0 PACKAGING



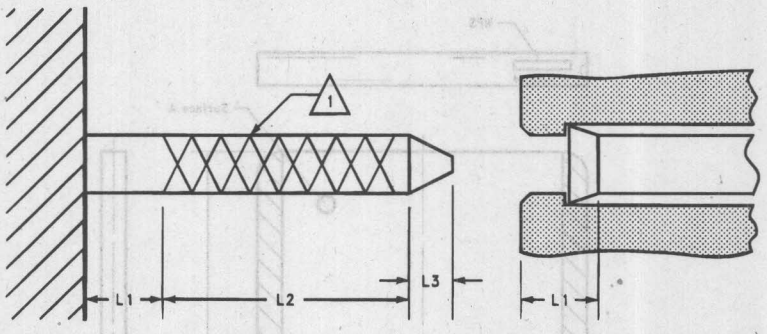
C MIN	L ± 0.008	P MIN 1	S MIN	T 2	W ± 0.004	X ± 0.002	Y ± 0.002
0.294 (10.0)	3.370 (85.60)	0.394 (10.0)	0.118 (3.0)	0.065 (1.65)	2.126 (54.0)	0.039 (1.00)	0.063 (1.60)

1 POLARIZATION KEY LENGTH.

2 INTERCONNECT AREA TOLERANCE = ±0.002
SUBSTRATE AREA TOLERANCE = ±0.004

3 MILLIMETERS ARE IN PARENTHESIS ().

049106.eps



1. PIN/SOCKET CONTACT AREA
2. MILLIMETERS ARE IN PARENTHESIS ()

049107.eps

14.0 ADDITIONAL INFORMATION

14.1 Ordering Information

iMC020FLSP,SBXXXXX

WHERE:

i	= INTEL
MC	= MEMORY CARD
020	= DENSITY IN MEGABYTES (004,020 AVAILABLE)
FL	= FLASH TECHNOLOGY
S	= BLOCKED ARCHITECTURE
P	= PERFORMANCE
SBXXXXX	= CUSTOMER IDENTIFIER

14.2 References

Order Number	Document
290434	Series 2+ Flash Memory Card Datasheet
297373	"Series 2+ Flash Memory Card User's Manual"
290489	"28F016SA 16-Mbit (1 Mb x 16, 2 Mb x 8) FlashFile™ Memory" Datasheet
290429	"28F08SA 8-Mbit (1 Mb x 8) FlashFile™ Memory" Datasheet
292092	AP-357, "Power Supply Solutions for Flash Memory"
292097	AP-362, "Implementing Mobile PC Designs Using High Density FlashFile™ Components"
292126	AP-377, "The 28F016SA Software Drivers"
292127	AP-378, "Enhanced Features of the 28F016SA"
292158	AP-606, "Interchangablility of Series 1, Series 2, and Series 2+ Flash Memory Cards"
294015	ER-31, "The 28F016SA FlashFile™ Memory"
294016	ER-33, "ETOX™ IV Flash Memory Technology: Insight to Intel's Fourth Generation Process Innovation"

14.3 Revision History

Number	Description
001	Original Version
002	Page Buffer Write to Flash Command Code Correction (from 08H to 0CH). Series 2+ Tuples and AC Characteristics Tables Include Support for 150 ns Access.
003	TTL DC Characteristics Tables Added. General DC Characteristics Table Changed to Reflect TTL Levels.
004	AC Characteristics Condensed to Include V_{PP} Pump Information.
005	Series 2+ 8-Meg and 40-Meg Cards Added to Datasheet. Table of Contents Numbering System Added. Component Management Register Tables Reformatted. I_{CCR} Values Increased for CMOS and TTL. 3.3V Timings Updated to 250 ns.
006	Changed I_{PPSL} and I_{PPS} Max Values.

SERIES 2 FLASH MEMORY CARDS
iMC002FLSA, iMC004FLSA, iMC010FLSA, iMC020FLSA

Extended Temperature Specifications Included

- **2, 4, 10 and 20 Megabyte Capacities**
- **PCMCIA 2.1/JEIDA 4.1 68-Pin Standard**
  - **— Hardwired Card Information Structure**
  - **— Byte- or Word-Wide Selectable**
- **Component Management Registers for Card Status/Control and Flexible System Interface**
- **Automatic Erase/Write**
  - **— Monitored with Ready/Busy Output**
- **Card Power-Down Modes**
  - **— Deep-Sleep for Low Power Applications**
- **Mechanical Write Protect Switch**
- **Solid-State Reliability**
- **Intel FlashFile™ Architecture**
- **High-Performance Read Access**
  - **— 150 ns Maximum**
- **High-Performance Random Writes**
  - **— 6 μs Typical Word Write**
- **Erase Suspend to Read Command**
  - **— Keeps Erase as Background Task**
- **Nonvolatility (Zero Retention Power)**
  - **— No Batteries Required for Back-up**
- **ETOX™ V 0.4μ Flash Memory Technology**
  - **— 5V Read, 12V Erase/Write**
  - **— High-Volume Manufacturing Experience**
- **Extended Temperature Version**
  - **— −40°C to +85°C**

Intel's Series 2 Flash Memory Card facilitates high-performance disk emulation in mobile PCs and dedicated equipment. Manufactured with Intel's ETOX™ III 0.8μ, FlashFile Memory devices, the Series 2 Card allows code and data retention while erasing and/or writing other blocks. Additionally, the Series 2 Flash Memory Card features low power modes, flexible system interfacing and a 150 ns read access time. When coupled with Intel's low-power microprocessors, these cards enable high-performance implementations of mobile computers and systems.

Series 2 Cards conform to the Personal Computer Memory Card International Association (PCMCIA 2.1)/Japanese Electronics Industry Development Association (JEIDA 4.1) 68-pin standard, providing electrical and physical compatibility.

Data file management software, Flash Translation Layer (FTL), provides data file storage and memory management, much like a disk operating system. Intel's Series 2 Flash Memory Cards, coupled with flash file management software, effectively provide a removable, all-silicon mass storage solution with higher performance and reliability than disk-based memory architectures.

Designing with Intel's FlashFile Architecture enables OEM system manufacturers to design and manufacture a new generation of mobile PCs and dedicated equipment where high performance, ruggedness, long battery life and lighter weight are a requirement. For large user groups in workstation environments, the Series 2 Cards provide a means to securely store user data and backup system configuration/status information.

Table 1. Series 2 Flash Memory Card Pinout

Pin	Signal	I/O	Function	Active
1	GND		Ground	
2	DQ_3	I/O	Data Bit 3	
3	DQ_4	I/O	Data Bit 4	
4	DQ_5	I/O	Data Bit 5	
5	DQ_6	I/O	Data Bit 6	
6	DQ_7	I/O	Data Bit 7	
7	CE_1#	I	Card Enable 1	LO
8	A_{10}	I	Address Bit 10	
9	OE#	I	Output Enable	LO
10	A_{11}	I	Address Bit 11	
11	A_9	I	Address Bit 9	
12	A_8	I	Address Bit 8	
13	A_{13}	I	Address Bit 13	
14	A_{14}	I	Address Bit 14	
15	WE#	I	Write Enable	LO
16	RDY/BSY#		Ready-Busy	HI/LO
17	V_{CC}		Supply Voltage	
18	V_{PP1}		Supply Voltage	
19	A_{16}	I	Address Bit 16	
20	A_{15}	I	Address Bit 15	
21	A_{12}	I	Address Bit 12	
22	A_7	I	Address Bit 7	
23	A_6	I	Address Bit 6	
24	A_5	I	Address Bit 5	
25	A_4	I	Address Bit 4	
26	A_3	I	Address Bit 3	
27	A_2	I	Address Bit 2	
28	A_1	I	Address Bit 1	
29	A_0	I	Address Bit 0	
30	DQ_0	I/O	Data Bit 0	
31	DQ_1	I/O	Data Bit 1	
32	DQ_2	I/O	Data Bit 2	
33	WP	O	Write Protect	HI
34	GND		Ground	

Pin	Signal	I/O	Function	Active
35	GND		Ground	
36	CD_1#	O	Card Detect 1	LO
37	DQ_{11}	I/O	Data Bit 11	
38	DQ_{12}	I/O	Data Bit 12	
39	DQ_{13}	I/O	Data Bit 13	
40	DQ_{14}	I/O	Data Bit 14	
41	DQ_{15}	I/O	Data Bit 15	
42	CE_2#	I	Card Enable 2	LO
43	VS_1	O	Voltage Sense 1	N.C.
44	RFU		Reserved	
45	RFU		Reserved	
46	A_{17}	I	Address Bit 17	
47	A_{18}	I	Address Bit 18	
48	A_{19}	I	Address Bit 19	
49	A_{20}	I	Address Bit 20	
50	A_{21}	I	Address Bit 21	
51	V_{CC}		Supply Voltage	
52	V_{PP2}		Supply Voltage	
53	A_{22}	I	Address Bit 22	
54	A_{23}	I	Address Bit 23	
55	A_{24}	I	Address Bit 24	
56	A_{25}		No Connect	
57	VS_2	O	Voltage Sense 2	N.C.
58	RST	I	Reset	HI
59	WAIT#	O	Extend Bus Cycle	LO
60	RFU		Reserved	
61	REG#	I	Register Select	LO
62	BVD_2	O	Batt. Volt Det 2	
63	BVD_1	O	Batt. Volt Det 1	
64	DQ_8	I/O	Data Bit 8	
65	DQ_9	I/O	Data Bit 9	
66	DQ_{10}	I/O	Data Bit 10	
67	CD_2#	O	Card Detect 2	LO
68	GND		Ground	

Table 2. Series 2 Flash Memory Card Pin Descriptions

Symbol	Type	Name and Function
A_0–A_{25}	I	**ADDRESS INPUTS:** A_0 through A_{25} are address bus lines which enable direct addressing of 64 megabytes of memory on a card. A_0 is not used in word access mode. A_{24} is the most significant address bit. Note: A_{25} is a no-connect but should be provided on host side.
DQ_0–DQ_{15}	I/O	**DATA INPUT/OUTPUT:** DQ_0 through DQ_{15} constitute the bidirectional data bus. DQ_{15} is the most significant bit.
CE_1#, CE_2#	I	**CARD ENABLE 1, 2:** CE_1# enables even bytes, CE_2# enables odd bytes. Multiplexing A_0, CE_1# and CE_2# allows 8-bit hosts to access all data on DQ_0 through DQ7. (See Table 3 for a more detailed description.)
OE#	I	**OUTPUT ENABLE:** Active low signal gating read data from the memory card.
WE#	I	**WRITE ENABLE:** Active low signal gating write data to the memory card.
RDY/BSY#	O	**READY/BUSY OUTPUT:** Indicates status of internally timed erase or write activities. A high output indicates the memory card is ready to accept accesses. A low output indicates that a device(s) in the memory card is(are) busy with internally timed activities. See text for an alternate function (READY-BUSY MODE REGISTER).
CD_1# & CD_2#	O	**CARD DETECT 1, 2:** These signals provide for correct card insertion detection. They are positioned at opposite ends of the card to detect proper alignment. The signals are connected to ground internally on the memory card and will be forced low whenever a card is placed in the socket. The host socket interface circuitry shall supply 10K or larger pull-up resistors on these signal pins.
WP	O	**WRITE PROTECT:** Write Protect reflects the status of the Write-Protect switch on the memory card. WP set high = write protected, providing internal hardware write lockout to the flash array.
V_{PP1}, V_{PP2}		**WRITE/ERASE POWER SUPPLY:** (12V nominal) for erasing memory array blocks or writing data in the array. They must be 12V to perform an erase/write operation.
V_{CC}		**CARD POWER SUPPLY** (5V nominal) for all internal circuitry.
GND	I	**GROUND** for all internal circuitry.
REG#	I	**REGISTER SELECT** provides access to Series 2 Flash Memory Card registers and Card Information Structure in the Attribute Memory Plane.
RST	I	**RESET** from system, active high. Places card in Power-On Default State. RESET pulse width must be ≥ 200 ns.
WAIT#	O	**WAIT (Extend Bus Cycle)** is used by Intel's I/O cards and is driven high.
BVD_1, BVD_2	O	**BATTERY VOLTAGE DETECT:** Upon completion of the power on reset cycle, these signals are driven high to maintain SRAM-card compatibility.
RFU		**RESERVED FOR FUTURE USE**
NC		**NO INTERNAL CONNECTION**. Pin may be driven or left floating.
VS_1, VS_2		**VOLTAGE SENSE:** Notifies the host Socket of the card's V_{CC} requirements. VS_1 and VS_2 are both open, indicating a 5V V_{CC} card.

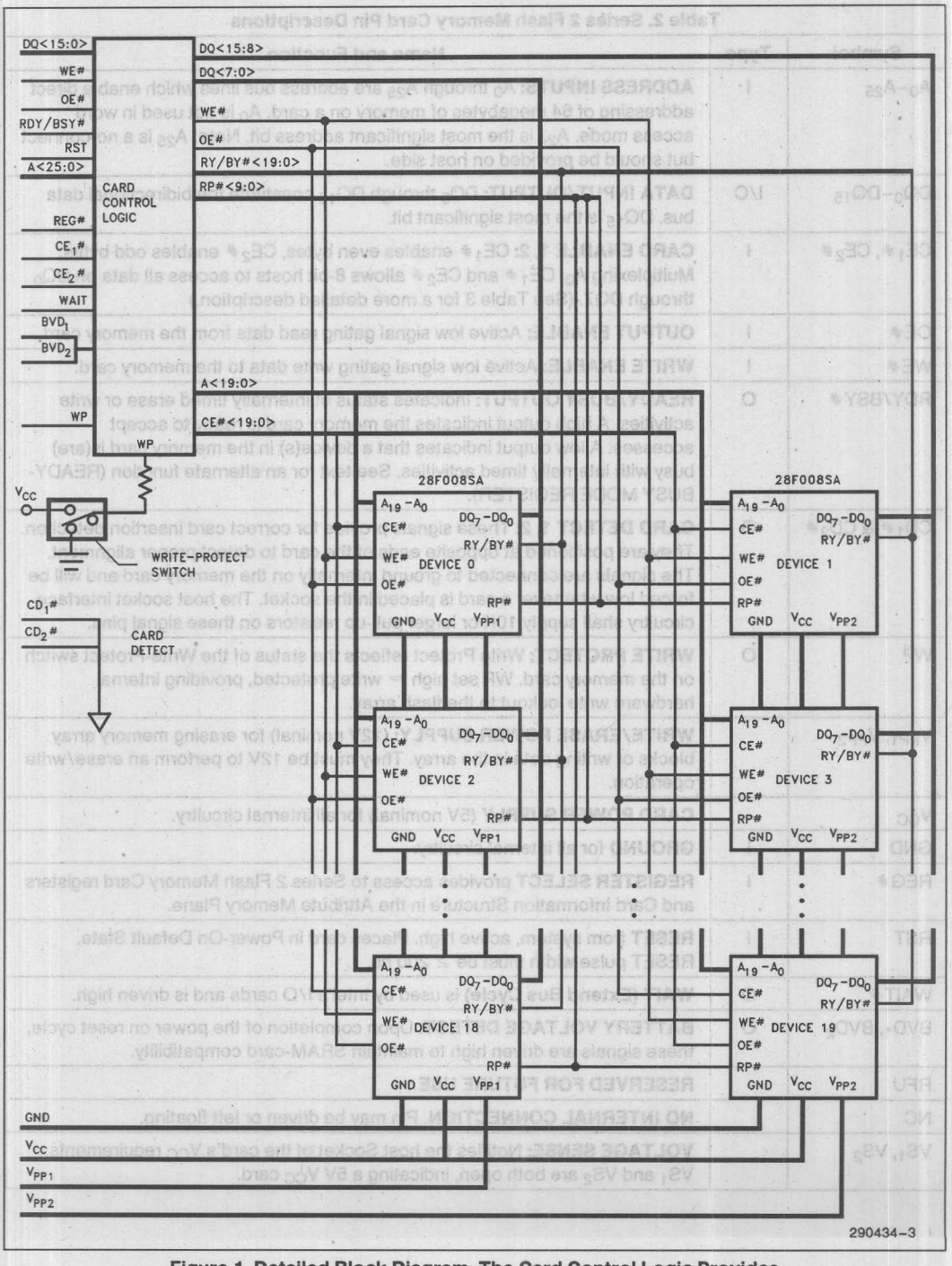


Figure 1. Detailed Block Diagram. The Card Control Logic Provides Decoding Buffering and Control Signals.

APPLICATIONS

Intel's second generation Series 2 Flash Memory Cards facilitate high performance disk emulation for the storage of data files and application programs on a purely solid-state removable medium. File management software, Flash Translation Layer (FTL), in conjunction with the Series 2 Flash Memory Cards, enable the design of high-performance light-weight notebook, palmtop, and pen-based PCs that have the processing power of today's desktop computers.

Application software stored on the flash memory card substantially reduces the slow disk-to-DRAM download process. Replacing the mechanical disk results in a dramatic enhancement of read performance and substantial reduction of power consumption, size and weight—considerations particularly important in portable PCs and equipment. The Series 2 Card's high performance read access time allows the use of Series 2 Cards in an "execute-in-place" (XIP) architecture. XIP eliminates redundancy associated with DRAM/Disk memory system architectures. Operating systems stored in Flash Memory decreases system boot or program load times, enabling the design of PCs that boot, operate, store data files and execute application programs from/to nonvolatile memory without losing the ability to perform an update.

File management systems modify and store data files by allocating flash memory space intelligently. Wear leveling algorithms, employed to equally distribute the number of rewrite cycles, ensure that no particular block is cycled excessively relative to other blocks. This provides hundreds of thousands of hours of power on usage.

This file management software enables the user to interact with the flash memory card in precisely the same way as a magnetic disk.

Series 2 Flash Memory Cards provide durable nonvolatile memory storage for mobile PCs on the road, facilitating simple transfer back into the desktop environment.

For systems currently using a static RAM/battery configuration for data acquisition, the Series 2 Flash Memory Card's nonvolatility eliminates the need for battery backup. The concern for battery failure no longer exists, an important consideration for portable computers and medical instruments, both requiring continuous operation. Series 2 Cards consume no power when the system is off, and only 60 μA in Deep-Sleep mode (2 Megabyte card). Furthermore, Flash Memory Cards offer a considerable cost and density advantage over memory cards based on static RAM with battery backup.

Besides disk emulation, the Series 2 Card's electrical block-erasure, data writability, and inherent nonvolatility fit well with data accumulation and recording needs. Electrical block-erasure provides design flexibility to selectively rewrite blocks of data, while saving other blocks for infrequently updated parameters and lookup tables. For example, networks and systems that utilize large banks of battery-backed DRAM to store configuration and status benefit from the Series 2 Flash Card's nonvolatility and reliability.

SERIES 2 ARCHITECTURE OVERVIEW

The Series 2 Flash Memory Card contains a 2 to 20 Megabyte Flash Memory array consisting of 2 to 20 28F008SA FlashFile Memory devices. Each 28F008SA contains sixteen individually-erasable, 64 Kbyte blocks; therefore, the Flash Memory Card contains from 32 to 320 device blocks. It also contains two Card Control Logic devices that manage the external interface, address decoding, and component management logic. (Refer to Figure 1 for a block diagram.)

To support PCMCIA-compatible word-wide access, devices are paired so that each accessible memory block is 64 KWords (see Figure 2). Card logic allows the system to write or read one word at a time, or one byte at a time by referencing the high or low byte. Erasure can be performed on the entire block pair (high and low device blocks simultaneously), or on the high or low byte portion separately.

Also in accordance with PCMCIA specifications this product supports byte-wide operation, in which the flash array is divided into 128K x 8 bit device blocks. In this configuration, odd bytes are multiplexed onto the low byte data bus.

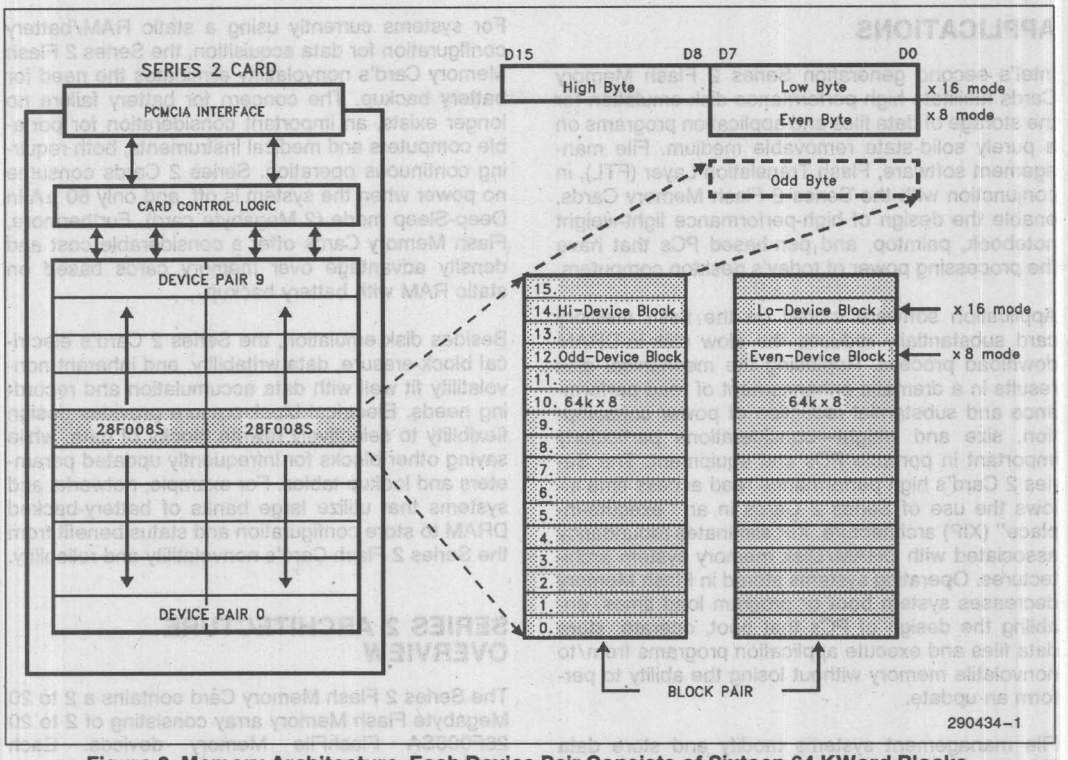


Figure 2. Memory Architecture. Each Device Pair Consists of Sixteen 64 KWord Blocks.

Series 2 Flash Memory Cards offer additional features over the Bulk Erase Flash Card product family (refer to iMC001FLKA, iMC002FLKA and iMC004FLKA data sheets). Some of the more notable enhancements include: high density capability, erase blocking, internal write/erase automation, erase suspension to read, Component Management Registers that provide software control of device-level functions and a deep-sleep mode.

Erase blocking facilitates solid-state storage applications by allowing selective memory reclamation. Multiple 64 Kbyte blocks may be simultaneously erased within the memory card as long as not more than one block per device is erasing. This shortens the total time required for erasure, but requires additional supply current. A block typically requires 1.6 seconds to erase. Each memory block can be erased and completely written 100,000 times.

Erase suspend allows the system to temporarily interrupt a block erase operation. This mode permits reads from alternate device blocks while that same device contains an erasing block. Upon completion of the read operation, erasure of the suspended block must be resumed.

Write/erase automation simplifies the system software interface to the card. A two-step command sequence initiates write or erase operations and provides additional data security. Internal device circuits automatically execute the algorithms and timings necessary for data-write or block-erase operations, including verifications for long-term data integrity. While performing either data-write or block-erase, the memory card interface reflects this by bringing its RDY/BSY# (Ready/Busy) pin low. This output goes high when the operation completes. This feature reduces CPU overhead and allows software polling or hardware interrupt mechanisms. Writing memory data is achieved in single byte or word increments, typically in 6 μs.

Read access time is 150 ns or less over the entire operating temperature range.

The Reset-PowerDown mode reduces power consumption to less than 60 μA to help extend battery life of portable host systems. Activated through software control, this mode optionally affects the entire flash array (Global Reset-PowerDown Register) or specific device pairs (Sleep Control Register).

PCMCIA/JEIDA INTERFACE

The Series 2 Flash Memory Card interface supports the PCMCIA 2.1 and JEIDA 4.1 68-pin card format (see Tables 1 and 2). Detailed specifications are described in the PC Card Standard, Release 2.1, July, 1993, published by PCMCIA. The Series 2 Card conforms to the requirements of both Release 1 and Release 2 of the PC Card Standard.

Series 2 Card pin definitions are equivalent to the Bulk-Erase Flash Card except that certain No Connects are now used. A_{22} through A_{24}, RST (Reset), and RDY/BSY# (Ready/Busy) have pin assignments as set by the PCMCIA standard.

NOTE: The READY/BUSY signal is abbreviated as RDY/BSY# by PCMCIA (card level) and as RY/BY# by JEDEC (component level).

The outer shell of the Series 2 card meets all PCMCIA/JEIDA Type 1 mechanical specifications. See Figure 19 for mechanical dimensions.

WRITE PROTECT SWITCH

A mechanical write protect switch provides the card's memory array with internal write lockout. The Write-Protect (WP) output pin reflects the status of this mechanical switch. It outputs a high signal (V_{OH}) when writes are disabled. This switch does not lock out writes to the Component Management Registers.

BATTERY VOLTAGE DETECT

PCMCIA requires two signals, BVD_1 and BVD_2, be supplied at the interface to reflect card battery condition. Flash Memory Cards do not require batteries. When the power on reset cycle is complete, BVD_1 and BVD_2 are driven high to maintain compatibility.

CARD DETECT

Two signals, CD_1# and CD_2#, allow the host to determine proper socket seating. They reside at opposite ends of the connector and are tied to ground within the memory card.

DESIGN CONSIDERATIONS

The Series 2 Card consists of two separate memory planes: the Common Memory Plane (or Main Memory) and the Attribute Memory Plane. The Common Memory Plane resides in the banks of device pairs and represents the user-alterable memory space.

The Component Management Registers (CMR) and the hardwired Card Information Structure (CIS) reside in the Attribute Memory Plane within the Card Control Logic, as shown in Figure 3. The Card Control Logic interfaces the PCMCIA connector and the internal flash memory array and performs address decoding and data control.

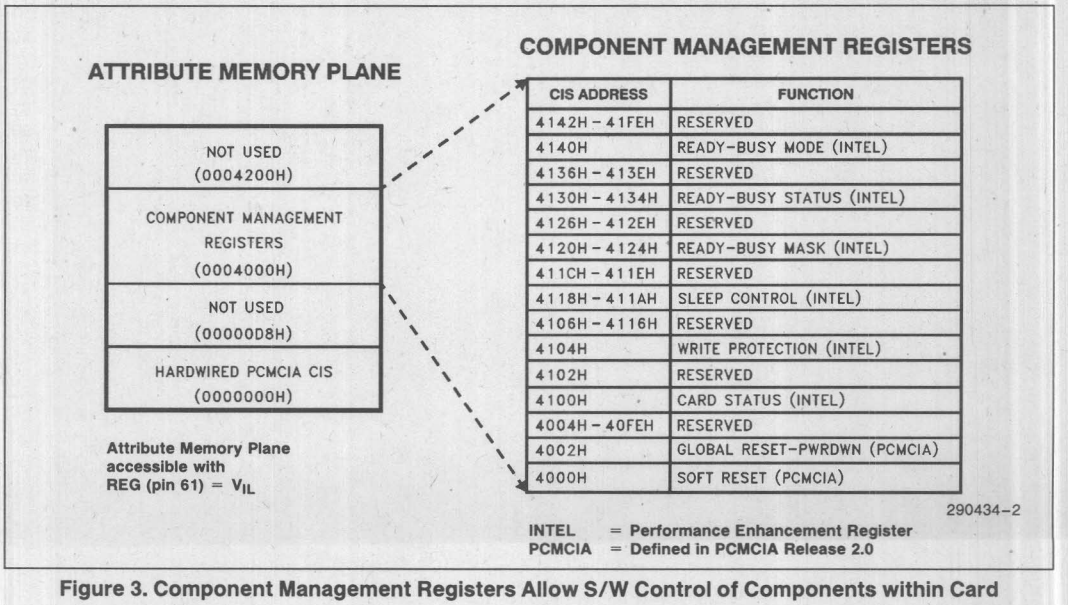


Figure 3. Component Management Registers Allow S/W Control of Components within Card

ADDRESS DECODE

Address decoding provides the decoding logic for the 2 to 20 Device Chip Enables and the elements of the Attribute Memory Plane. REG# selects between the Common Memory Plane (REG# = V_{IH}) and the Attribute Memory Plane (REG# = V_{IL}).

NOTE:
The Series 2 Card has *active* address inputs A_0 to A_{24} implying that reading and writing to addresses beyond 32 Megabytes causes wraparound. Furthermore, reads to illegal addresses (for example, between 20 and 32 Meg on a 20 Megabyte card) returns Default data (00FFH or FFFFH).

The 28F008SA devices, storing data, applications or firmware, form the Common Memory Plane accessed individually or as device pairs. Memory is linearly mapped in the Common Memory Plane. Three memory access modes are available when accessing the Common Memory Plane: Byte-Wide, Word Wide, and Odd-Byte modes.

Additional decoding selects the hardwired PCMCIA CIS and Component Management Registers mapped in the Attribute Memory Plane beginning at address 000000H.

The 512 memory-mapped even-byte CMRs are linearly mapped beginning at address 4000H in the Attribute Memory Plane.

DATA CONTROL

Data Control Logic selects the path and direction for accessing the Common or Attribute Memory Plane. It controls any of the PCMCIA-defined Word-Wide, Byte-Wide or Odd-Byte modes for either reads or writes to these areas. As shown in Table 3, input pins which determine these selections are REG#, A_0 through A_{24}, WE#, OE#, CE_1#, and CE_2#. PCMCIA specifications allow only even-byte access to the Attribute Memory Plane.

In Byte-Wide mode, bytes contiguous in software actually alternate between two device blocks of a device pair. Therefore, erasure of one device block erases every other contiguous byte. In accordance with the PCMCIA standard for memory configuration, the Series 2 Card does not support confining contiguous bytes within one flash device when in by-8 mode.

Table 3. Data Access Mode Truth Table

Function Mode	REG#	$CE\#_2$	$CE\#_1$	A_0	OE#	WE#	V_{PP2}	V_{PP1}	D_{15}–D_8	D_7–D_0
COMMON MEMORY PLANE										
STANDBY(1)	X	H	H	X	X	X	V_{PPL}(2)	V_{PPL}(2)	HIGH-Z	HIGH-Z
BYTE READ	H	H	L	L	L	H	V_{PPL}(2)	V_{PPL}(2)	HIGH-Z	EVEN-BYTE
	H	H	L	H	L	H	V_{PPL}(2)	V_{PPL}(2)	HIGH-Z	ODD-BYTE
WORD READ	H	L	L	X	L	H	V_{PPL}(2)	V_{PPL}(2)	ODD-BYTE	EVEN-BYTE
ODD-BYTE READ	H	L	H	X	L	H	V_{PPL}(2)	V_{PPL}(2)	ODD-BYTE	HIGH-Z
BYTE WRITE	H	H	L	L	H	L	V_{PPH}	V_{PPH}	X	EVEN-BYTE
	H	H	L	H	H	L	V_{PPH}	V_{PPH}	X	ODD-BYTE
WORD WRITE	H	L	L	X	H	L	V_{PPH}	V_{PPH}	ODD-BYTE	EVEN-BYTE
ODD-BYTE WRITE	H	L	H	X	H	L	V_{PPH}	V_{PPL}(2)	ODD-BYTE	X
ATTRIBUTE MEMORY PLANE										
BYTE READ	L	H	L	L	L	H	X(2)	X(2)	HIGH-Z	EVEN-BYTE
	L	H	L	H	L	H	X(2)	X(2)	HIGH-Z	INVALID
WORD READ	L	L	L	X	L	H	X(2)	X(2)	INVALID DATA(3)	EVEN-BYTE
ODD-BYTE READ	L	L	H	X	L	H	X(2)	X(2)	INVALID DATA(3)	HIGH-Z
BYTE WRITE	L	H	L	L	H	L	X(2)	X(2)	X	EVEN-BYTE
	L	H	L	H	H	L	X(2)	X(2)	X	INVALID OPERATION(3)
WORD WRITE	L	L	L	X	H	L	X(2)	X(2)	INVALID OPERATION(3)	EVEN-BYTE
ODD-BYTE WRITE	L	L	H	X	H	L	X(2)	X(2)	INVALID OPERATION(3)	X

NOTES:
1. Standby mode is valid in Common Memory or Attribute Memory access.
2. To meet the low power specifications, $V_{PP} = V_{PPL}$; however V_{PPH} presents no reliability problems.
3. Odd-Byte data are not valid during access to the Attribute Memory Plane.
4. H = V_{IH}, L = V_{IL}, X = Don't Care.

PRINCIPLES OF OPERATION

Intel's Series 2 Flash Memory Card provides electrically-alterable, non-volatile, random-access storage. Individual 28F008SA devices utilize a Command User Interface (CUI) and Write State Machine (WSM) to simplify block-erasure and data write operations.

COMMON MEMORY ARRAY

Figure 4 shows the Common Memory Plane's organization. The first block pair (64 KWords) of Common Memory, referred to as the Common Memory Card Information Structure Block, *optionally* extends the hardwired CIS in the Attribute Memory Plane for additional card information. This may be written during initial card formatting for OEM customization. Since this CIS Block is part of Common Memory, its data can be altered. Write access to the Common Memory CIS Block is controlled by the Write Protect Control Register which may be activated by system software after power-up. Additionally, the entire Common Memory plane (minus the Common Memory CIS Block) may be software write protected. *Note that the Common Memory CIS Block is not part of the Attribute Memory Plane. Do not assert REG# to access the Common Memory CIS Block.*

13FFFFFH 1200000H	Device Pair 9
1000000H	Device Pair 8
0E00000H	Device Pair 7
0C00000H	Device Pair 6
0A00000H	Device Pair 5
0800000H	Device Pair 4
0600000H	Device Pair 3
0400000H	Device Pair 2
0200000H	Device Pair 1
0020000H	Device Pair 0
0000000H	Optional CIS

Figure 4. Common Memory Plane. Use the Optional Common Memory Plane CIS for Custom Card Format Information.

HARDWIRED CIS

The card's structure description resides in the even-byte locations starting at 0000H and going to the CIS ending tuple (FNULL) within the Attribute Memory Plane. Data included in the hardwired CIS consists of tuples. Tuples are a variable-length list of data blocks describing details such as manufacturer's name, the size of each memory device and the number of flash devices within the card.

COMPONENT MANAGEMENT REGISTERS (CMRs)

The CMRs in the Attribute Memory Plane provide special, software-controlled functionality. Card Control Logic includes circuitry to access the CMRs. REG (PCMCIA, pin 61) selects the Attribute Memory Plane (and therefore the CMRs) when equal to V_{IL}.

CMRs are classified into two categories: those defined by PCMCIA R2.1 and those included by Intel (referred to as Performance Enhancement Registers) to enhance the interface between the host system and the card's flash memory array. CMRs (See Figure 3) provide seven control functions—Ready-Busy Interrupt Mode, Device Ready-Busy Status, Device Ready-Busy Mask, Reset-PowerDown Control, Software-controlled Write Protection, Card Status and Soft Reset.

SOFT RESET REGISTER (PCMCIA)
(CONFIGURATION OPTION)

The SOFT RESET REGISTER (Attribute Memory Plane Address 4000H, Figure 5) is defined in the PCMCIA Release 2.0 specification as the Configuration Option Register.

Bit 7 is the soft reset bit (SRESET). Writing a 1 to this bit initiates card reset to the power-on default state (see Side Bar page 11). This bit must be cleared to use the CMRs or to access the devices.

SRESET implements in software what the reset pin implements in hardware. On power-up, the card automatically assumes default conditions. Similar to the reset pin (pin 58), this bit clears at the end of a power-on reset cycle or a system reset cycle.

Bits 0 through 6 are not used by this memory card, but power up as zeroes for PCMCIA compatibility.

SOFT RESET REGISTER

(CONFIGURATION OPTION REGISTER)

(Read/Write Register)

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 4	BIT 2	BIT 1	BIT 0
4000H	SRESET	PCMCIA CONFIGURATION INDEX RESETS TO ZERO ON POWER UP						

1 = RESET, CLEAR TO ACCESS CARD

Figure 5. SOFT RESET REGISTER (PCMCIA). Sets the Memory Card in the Power-On Default State.

POWER-ON DEFAULT CONDITIONS

- All Devices Powered Up In Standby Mode
- Common Memory Available For Writes
- All Device Ready/Busy Outputs Unmasked
- PCMCIA Ready/Busy Mode Enabled
- Ready/Busy Output Goes To Ready

Global PowerDown Register (PCMCIA)

(Configuration and Status)

The Global Reset-PowerDown Register (Attribute Memory Plane Address 4002H, Figure 6) is referred to as the Configuration and Status Register in the PCMCIA Release 2.0 specification.

Bit 2 (RP) controls global card power-down. Writing a 1 to this bit places each device within the card into "Deep-Sleep" mode. *Devices in Deep-Sleep are not accessible.* Recovery from power-down requires 500 ns for reads and 1 μs for writes.

The RP bit defaults to 0 on card power-up or reset. Setting or clearing this bit has no affect on the bit settings of the Sleep Control Register.

The remaining Global Reset-PwrDwn Register bits are defined for Intel's family of I/O cards and are driven low for compatibility.

GLOBAL RESET-POWER-DOWN REGISTER

(CONFIGURATION AND STATUS REGISTER)

(Read/Write Register)

1 = POWER DOWN

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
4002H	ZEROES					RP	ZEROES	

Figure 6. GLOBAL RESET-PWRDWN REGISTER (PCMCIA). The RP Bit Enables Reset PowerDown of All Flash Memory Devices.

CARD STATUS REGISTER

(Read Only Register)

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
4100H	ADM	ADS	SRESET	CMWP	RP	CISWP	WP	RDY/BSY#

Figure 7. CARD STATUS REGISTER (Intel) Provides a Quick Review of the Card's Status

CARD STATUS REGISTER (INTEL)

The Read-Only, CARD STATUS REGISTER (Attribute Memory Plane Address 4100H, Figure 7) returns generalized status of the Series 2 Card and its CMRs.

Bit 0 (RDY/BSY#) reflects the card's RDY/BSY# (Ready-Busy) output. Software polling of this bit provides data-write or block-erase operation status. A zero indicates a busy device(s) in the card.

Bit 1 (WP) reports the position of the card's Write Protection switch with 1 indicating write protected. It reports the status of the WP pin.

Bit 2 (CISWP) reflects whether the Common Memory CIS is write protected using the WRITE PROTECT REGISTER, with 1 indicating write protected.

Bit 3 (RP) reports whether the entire flash memory array is in "Deep-Sleep" (Reset-PwrDwn) mode, with 1 indicating "Deep-Sleep". This bit reflects the RP bit of the GLOBAL RESET-POWERDOWN REGISTER. Powering down *all* device pairs individually (using the Sleep Control Register), also sets this bit.

Bit 4 (CMWP) reports whether the Common Memory Plane (minus Common Memory CIS) is write protected via the WRITE PROTECT REGISTER with 1 indicating write protected.

Bit 5 (SRESET) reflects the SRESET bit of the SOFT RESET REGISTER. It reports that the card is in Soft Reset with 1 indicating reset. When this bit is zero, the flash memory array and CMRs may be accessed, otherwise clear it via the SRESET REGISTER.

Bit 6 (ADS, ANY DEVICE SLEEP) is the "ORed" value of the SLEEP CONTROL REGISTER. Powering down any device pair sets this bit.

Bit 7 (ADM, ANY DEVICE MASKED) is the "ORed" value of the READY/BUSY MASK REGISTER. Masking any device sets this bit.

WRITE PROTECTION REGISTER (INTEL)

The WRITE PROTECTION REGISTER (Attribute Memory Plane Address 4104H, Figure 8) selects whether the optional Common Memory CIS and the remaining Common Memory blocks are write protected (see Figure 4).

Enable Common Memory CIS write protection by writing a 1 to the CISWP Bit (bit 0).

Enable write protection of the remaining Common Memory blocks by writing a 1 to the CMWP Bit (bit 1).

In the power-on default state, both bits are 0, and therefore not write protected.

Reserved bits (2–7) have undefined values and should be written as zeroes for future compatibility.

WRITE PROTECTION REGISTER

(Read/Write Register)

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
4104H	RESERVED FOR FUTURE USE						CMWP	CISWP

1 = WRITE PROTECT

Figure 8. WRITE PROTECTION REGISTER (Intel) Eliminates Accidental Data Corruption

SLEEP CONTROL REGISTER (INTEL)

Unlike the GLOBAL RESET-POWERDOWN REGISTER, which simultaneously resets and places all flash memory devices into a Deep-Sleep mode, the SLEEP CONTROL REGISTER (Attribute Memory Plane Address 4118H–411AH, Figure 9) allows selective power-down control of individual device pairs.

Writing a 1 to a specific bit of the SLEEP CONTROL REGISTER places the corresponding device pair into the "Deep-Sleep" mode. *Devices in Deep-Sleep are not accessible*. On cards with fewer than 20 Megabytes (10 device pairs), writing a one to an absent device pair has no affect and reads back as zero.

This register contains all zeroes (i.e., not in Deep-Sleep mode) when the card powers up or after a hard or soft reset. Furthermore, the Global Reset-PowerDown Register has no affect on the contents of this register. Therefore, any bit settings of the Sleep Control Register will remain unchanged after returning from a global reset and power down (writing a zero to the RP bit of the Global Reset-Power-Down Register).

READY-BUSY STATUS REGISTER (INTEL)

The bits in the Read-only, READY-BUSY Status Register (Attribute Memory Plane Address 4130H-4134H, Figure 10) reflect the status (READY = 1, BUSY = 0) of each device's RY/BY# output. A busy condition indicates that a device is currently processing a data-write or block-erase operation.

These bits are logically "AND-ed" to form the Ready/Busy output (RDY/BSY#, pin 16) of the PCMCIA interface. On memory cards with fewer than 20 devices, unused Device RY/BY# Status Register bits appear as ready.

SLEEP CONTROL REGISTER

(Read/Write Register)

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
411AH	RESERVED						DEVICES 18/19	DEVICES 16/17
4118H	DEVICES 14/15	DEVICES 12/13	DEVICES 10/11	DEVICES 8/9	DEVICES 6/7	DEVICES 4/5	DEVICES 2/3	DEVICES 0/1

1 = SELECTED DEVICE PAIR IN POWER-DOWN MODE AND RESET

Figure 9. SLEEP CONTROL REGISTER (Intel) Allows Specific Devices to be Reset and Put into Power-Down Mode

READY-BUSY STATUS REGISTER

(Read/Write Register)

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
4134H	RESERVED				DEVICE 19	DEVICE 18	DEVICE 17	DEVICE 16
4132H	DEVICE 15	DEVICE 14	DEVICE 13	DEVICE 12	DEVICE 11	DEVICE 10	DEVICE 9	DEVICE 8
4130H	DEVICE 7	DEVICE 6	DEVICE 5	DEVICE 4	DEVICE 3	DEVICE 2	DEVICE 1	DEVICE 0

1 = DEVICE READY, 0 = DEVICE BUSY

Figure 10. READY-BUSY STATUS REGISTER (Intel) Provides Operation Status of All Flash Memory Devices

READY-BUSY MASK REGISTER (INTEL)

The bits of the Read/Write READY-BUSY MASK REGISTER (Attribute Memory Plane Address 4120H–4124H, Figure 11) mask out the corresponding "AND-ed" READY-BUSY STATUS REGISTER bits from the PCMCIA data bus (RDY/BSY#, pin 16) and the CARD STATUS REGISTER RDY/BSY# Bit (bit 0).

In an unmasked condition (MASK REGISTER bits = 0), any device RY/BY# output going low pulls the card's RDY/BSY# output to V_{IL} (BUSY). In this case, all devices must be READY to allow the card's RDY/BSY# output to be ready (V_{IH}). *This is referred to as the PCMCIA READY-BUSY MODE. An alternate type of READY-BUSY function is described in the next section, READY-BUSY MODE REGISTER.*

READY-BUSY MASK

(Read/Write Register)

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
4124H	RESERVED				DEVICE 19	DEVICE 18	DEVICE 17	DEVICE 16
4122H	DEVICE 15	DEVICE 14	DEVICE 13	DEVICE 12	DEVICE 11	DEVICE 10	DEVICE 9	DEVICE 8
4120H	DEVICE 7	DEVICE 6	DEVICE 5	DEVICE 4	DEVICE 3	DEVICE 2	DEVICE 1	DEVICE 0

1 = MASK ENABLED

Figure 11. READY-BUSY MASK REGISTER (Intel) Essential for Write Optimization

If the READY-BUSY MASK REGISTER bits are set to ones (masked condition), the RDY/BSY# output and the CARD STATUS REGISTER RDY/BSY# bit will reflect a READY condition regardless of the state of the corresponding devices. The READY-BUSY MASK REGISTER does not affect the READY-BUSY STATUS REGISTER allowing software polling to determine operation status.

Unmasked is the default condition for the bits in this register. On memory cards with fewer than 20 devices, unused device mask bits appear as masked.

READY-BUSY MODE REGISTER (INTEL)

The READY-BUSY MODE REGISTER (Attribute Memory Plane Address 4140H, Figure 12) provides the selection of two types of system interfacing for the busy-to-ready transition of the card's RDY/BSY# pin:

1. The standard PCMCIA READY-BUSY MODE, in which the card's RDY/BSY# signal generates a low-to-high transition (from busy to ready) only after *all* busy devices (not including masked devices) have completed their data-write or block-erase operations. This may result in a long interrupt latency.
2. A High-Performance mode that generates a low-to-high (from busy-to-ready) transition after each device becomes ready. This provides the host system with immediate notification that a specific device's operation has completed and that device may now be used. This is particularly useful in a file management application where a block pair, containing only deleted files, is being erased to free up space so new file data may be written.

Enabling the HIGH-PERFORMANCE READY-BUSY MODE requires a three step sequence:

1. Set all bits in the READY/BUSY MASK REGISTER. This prevents ready devices from triggering an unwanted interrupt when step 3 is performed.
2. Write 01H to the READY-BUSY MODE REGISTER. This sets the MODE bit.
3. Write 01H to the READY-BUSY MODE REGISTER. This clears the RACK bit.

The MODE and RACK bits *must* be written in the prescribed sequence, *not* simultaneously. The card's circuitry is designed purposely in this manner to prevent an initial, unwanted busy-to-ready transition. Note that in Step 2, writing to the RACK bit is a Don't Care.

When the High-Performance Mode is enabled, specific READY-BUSY MASK bits must be cleared after an operation is initiated on the respective devices. After each device becomes ready, the RDY/BSY# pin makes a low-to-high transition. To catch the next device's completion of an operation, the RACK bit must be cleared by writing "01H" to the Ready/Busy Mode Register.

READY-BUSY MODE REGISTER

(Read/Write Register)

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
4140H	RESERVED FOR FUTURE USE						RACK	MODE

MODE = READY-BUSY MODE
0 = PCMCIA MODE
1 = HIGH PERFORMANCE

RACK = READY ACKNOWLEDGE CLEAR TO SET UP RDY/BSY# PIN, THEN CLEAR AFTER EACH DEVICE BECOMES READY TO ACKNOWLEDGE TRANSITION.

Figure 12. High Performance Ready-Busy Mode REGISTER (Intel) Used to Trigger a Ready Interrupt for Each Device

PRINCIPLES OF DEVICE OPERATION

Individual 28F008SA devices include a Command User Interface (CUI) and a Write State Machine (WSM) to manage write and erase functions in each device block.

The CUI serves as the device's interface to the Card Control Logic by directing commands to the appropriate device circuitry (Table 4). It allows for fixed power supplies during block erasure and data writes. The CUI handles the WE# interface into the device data and address latches, as well as system software requests for status while the WSM is operating.

The CUI itself does not occupy an addressable memory location. The CUI provides a latch used to store the command and address and data information needed to execute the command. Erase Setup and Erase Confirm commands require both appropriate command data and an address within the block to be erased. The Data Write Setup command requires both appropriate command data and the address of the location to be written, while the Data Write command consists of the data to be written and the address of the location to be written.

The CUI initiates flash memory writing and erasing operations only when V_{PP} is at 12V. Depending on the application, the system designer may choose to make the V_{PP} power supply switchable (available when writes and erases are required) or hardwired to V_{PPH}. When $V_{PP} = V_{PPL}$, power savings are incurred and memory contents cannot be altered. The CUI architecture provides protection from unwanted write and erase operations even when high voltage is applied to V_{PP}. Additionally, all functions are disabled whenever V_{CC} is below the write lockout voltage V_{LKO}, or when the card's Deep-Sleep modes are enabled. The WSM automates the writing and erasure of blocks within a device. This on-chip state machine controls block erase and data-write, freeing the host processor for other tasks. After receiving the Erase Setup and Erase Confirm commands from the CUI, the WSM controls block-erase. Progress is monitored via the device's status register, the card's control logic, and the RDY/BSY# pin of the PCMCIA interface. Data-write is similarly controlled, after destination address and expected data are supplied.

Table 4. Device Command Set

28F008SA Command[1]	Bus Cycles Req'd	First Bus Ccyle				Second Bus Cycle			
		Operation	Addr[2]	Data		Operation	Addr[2]	Data	
				x8 Mode	x16 Mode			x8 Mode	x16 Mode
Read Array/Reset	1	Write	DA	FFH	FFFFH				
Intelligent Identifer	3	Write	DA	90H	9090H	Read	IA	IID[3]	IID[3]
Read Device Status Register	2	Write	DA	70H	7070H	Read	DA	SRD[4]	SRD[4]
Clear Device Status Register	1	Write	DA	50H	5050H				
Erase Setup/Erase Confirm	2	Write	BA	20H	2020H	Write	BA	D0H	D0D0H
Erase Suspend/ Erase Resume	2	Write	DA	B0H	B0B0H	Write	DA	D0H	D0D0H
Write Setup/Write	2	Write	WA	40H	4040H	Write	WA	WD[5]	WD[5]
Alternate Write Setup/Write[6]	2	Write	WA	10H	1010H	Write	WA	WD[5]	WD[5]

NOTES:
1. Commands other than those shown above are reserved by Intel for future device implementations and should not be used.
2. DA = A device-level (or device pair) address within the card.
   BA = Address within the block of a specific device (device pair) being erased.
   WA = Address of memory location to be written.
   IA = A device-level address; 00H for manufacturer code, 01 for device code.
3. Following the intelligent identifier command, two read operations access manufacturer (89H) and device codes (A2H).
4. SRD = Data read from Device Status Register.
5. WD = Data to be written at location WA. Data is latched on the rising edge of WE#.
6. Either 40H or 10H are recognized by the WSM as the Write Setup command.

COMMAND DEFINITIONS

Read Array (FFH) —

Upon initial card power-up, after exit from the Deep-Sleep modes, and whenever illegal commands are given, individual devices default to the Read Array mode. This mode is also entered by writing FFH into the CUI. In this mode, microprocessor read cycles retrieve array data. Devices remain enabled for reads until the CUI receives an alternate command. Once the internal WSM has started a block-erase or data-write operation within a device, that device will not recognize the Read Array command until the WSM has completed its operation (or the Erase Suspend command is issued during erase).

Intelligent Identifier (90H) —

After executing this command, the intelligent identifier values can be read. Only address A_0 of each device is used in this mode, all other address inputs are reserved and should be cleared to 0. [(Manufacturer code = 89H for A_0 = 0), (Device code = A2H for A_0 = 1)]. The device will remain in this mode until the CUI receives another command.

This information is useful by system software in determining what type of flash memory device is contained within the card and allows the correct matching of device to write and erase algorithms. System software that fully utilizes the PCMCIA specification will not use the intelligent identifier mode, as this data is available within the Card Information Structure (refer to section on PCMCIA Card Information Structure).

Read Status Register (70H)

After writing this command, a device read outputs the contents of its Status Register, regardless of the address presented to that device. The contents of this register are latched on the falling edge of OE#, CE_1# (and/or CE_2#), whichever occurs last in the read cycle. This prevents possible bus errors which might occur if the contents of the Status Register changed while reading its contents. CE_1# (and CE_2# for odd-byte or word access) or OE# must be toggled with each subsequent status read, or the completion of a write or erase operation will not be evident. This command is executable while the WSM is operating, however, during a block-erase or data-write operation, reads from the device will automatically return status register data. Upon completion of that operation, the device remains in the Status Register read mode until the CUI receives another command.

The read Status Register command functions when $V_{PP} = V_{PPL}$ or V_{PPH}.

Clear Status Register (50H)

The Erase Status and Write Status bits may be set to "1"s by the WSM and can only be reset by the Clear Status Register Command. These bits indicate various failure conditions. By allowing system software to control the resetting of these bits, several operations may be performed (such as cumulatively writing several bytes or erasing multiple blocks in sequence). The device's Status Register may then be polled to determine if an error occurred during that sequence. This adds flexibility to the way the device may be used.

Additionally, the V_{PP} Status bit (SR.3) MUST be reset by system software (Clear Status Register command) before further block-erases are attempted (after an error).

The Clear Status Register command functions when $V_{PP} = V_{PPL}$ or V_{PPH}. This command puts the device in the Read Array mode.

Write Setup/Write

A two-command sequence executes a data-write operation. After the system switches V_{PP} to V_{PPH}, the write setup command (40H) is written to the CUI of the appropriate device, followed by a second write specifying the address and write data (latched on the rising edge of WE#). The device's WSM controls the data-write and write verify algorithms internally. After receiving the two-command write sequence, the device automatically outputs Status Register data when read (see Figure 13). The CPU detects the completion of the write operation by analyzing card-level or device-level indicators. Card-level indicators include the RDY/BSY# pin and the READY-BUSY STATUS REGISTER; while device-level indicators include the specific device's Status Register. Only the Read Status Register command is valid while the write operation is active. Upon completion of the data-write sequence (see section on Status Register) the device's Status Register reflects the result of the write operation. The device remains in the Read Status Register mode until the CUI receives an alternate command.

Erase Setup/Erase Confirm Commands (20H)

Within a device, a two-command sequence initiates an erase operation on one device block at a time. After the system switches V_{PP} to V_{PPH}, an Erase Setup command (20H) prepares the CUI for the Erase Confirm command (D0H). The device's WSM controls the erase algorithms internally. After receiving the two-command erase sequence, the device automatically outputs Status Register data when read (see Figure 14). If the command after erase setup is not an Erase Confirm command, the CR sets the Write Failure and Erase Failure bits of the Status Register, places the device into the Read Status Register mode, and waits for another command. The Erase Confirm command enables the WSM for erase (simultaneously closing the address latches for that device's block (A_{16}–A_{19}). The CPU detects the completion of the erase operation by analyzing card-level or device-level indicators. Card-level indicators include the RDY/BSY pin and the READY-BUSY STATUS REGISTER; while device-level indicators include the specific device's Status Register. Only the Read Status Register and Erase Suspend command is valid during an active erase operation. Upon completion of the erase sequence (see section on Status Register) the device's Status Register reflects the result of the erase operation. The device remains in the Read Status Register mode until the CUI receives an alternate command.

The two-step block-erase sequence ensures that memory contents are not accidentally erased. Erase attempts while $V_{PPL} < V_{PP} < V_{PPH}$ produce spurious results and are not recommended. Reliable block erasure only occurs when $V_{PP} = V_{PPH}$. In the absence of this voltage, memory contents are protected against erasure. If block erase is attempted while $V_{PP} = V_{PPL}$, the V_{PP} Status bit will be set to "1".

When erase completes, the Erase Status bit should be checked. If an erase error is detected, the device's Status Register should be cleared. The CUI remains in Read Status Register mode until receiving an alternate command.

Erase Suspend (B0H)/Erase Resume (D0H)

Erase Suspend allows block erase interruption to read data from another block of the device or to temporarily conserve power for another system operation. Once the erase process starts, writing the Erase Suspend command to the CUI (see Figure 15) requests the WSM to suspend the erase sequence at a predetermined point in the erase algorithm. In the erase suspend state, the device continues to output Status Register data when read.

Polling the device's RY/BY# and Erase Suspend Status bits (Status Register) will determine when the erase suspend mode is valid. It is important to note that the card's RDY/BSY# pin will also transition to V_{OH} and will generate an interrupt if this pin is connected to a system-level interrupt. At this point, a Read Array command can be written to the device's CUI to read data from blocks **other than those which are suspended**. The only other valid commands at this time are Read Status Register (70H) and Erase Resume (D0H). If V_{PP} goes low during Erase Suspend, the V_{PP} Status bit is set in the Status Register and the erase operation is aborted.

The Erase Resume command clears the Erase Suspend state and allows the WSM to continue with the erase operation. The device's RY/BY# Status and Erase Suspend Status bits and the card's READY-BUSY Status Register are automatically updated to reflect the erase resume condition. The card's RDY/BSY# pin also returns to V_{OL}.

Invalid/Reserved

These are unassigned commands having the same effect as the Read Array command. Do not issue any command other than the valid commands specified above. Intel reserves the right to redefine these codes for future functions.

DEVICE STATUS REGISTER

Each 28F008SA device in the Series 2 Card contains a Status Register which displays the condition of its Write State Machine. The Status Register is read at any time by writing the Read Status command to the CUI. After writing this command, all subsequent Read operations output data from the Status Register, until another command is written to the CUI.

Bit 7—WSM Status

This bit reflects the Ready/Busy condition of the WSM. A "1" indicates that read, block-erase or data- write operations are available. A "0" indicates that write or erase operations are in progress.

Bit 6—Erase Suspend Status

If an Erase Suspend command is issued during the erase operation, the WSM halts execution and sets the WSM Status bit and the Erase Suspend Status bit to a "1". This bit remains set until the device receives an Erase Resume command, at which point the CUI resets the WSM Status bit and the Erase Suspend Status bit.

Bit 5—Erase Status

This bit will be cleared to 0 to indicate a successful block-erasure. When set to a "1", the WSM has been unsuccessful at performing an erase verification. The device's CUI only resets this bit to a "0" in response to a Clear Status Register command.

Bit 4—Write Status

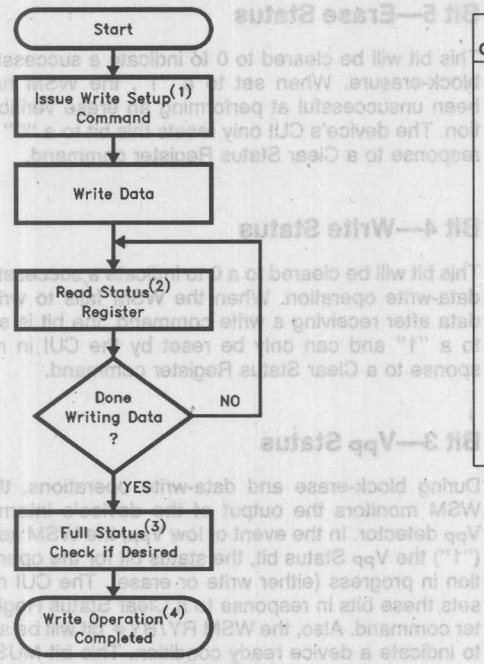
This bit will be cleared to a 0 to indicate a successful data-write operation. When the WSM fails to write data after receiving a write command, the bit is set to a "1" and can only be reset by the CUI in response to a Clear Status Register command.

Bit 3—V_{PP} Status

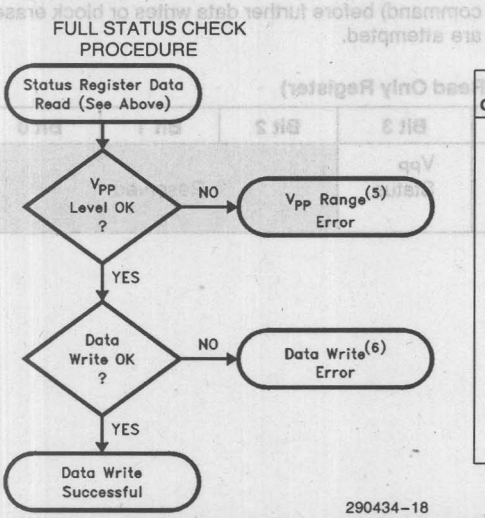
During block-erase and data-write operations, the WSM monitors the output of the device's internal V_{PP} detector. In the event of low V_{PP}, the WSM sets ("1") the V_{PP} Status bit, the status bit for the operation in progress (either write or erase). The CUI resets these bits in response to a Clear Status Register command. Also, the WSM RY/BY# bit will be set to indicate a device ready condition. This bit MUST be reset by system software (Clear Status Register command) before further data writes or block erases are attempted.

Device Status Register (Read Only Register)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WSM Status	Erase Suspend Status	Erase Status	Write Status	V_{PP} Status	Reserved		



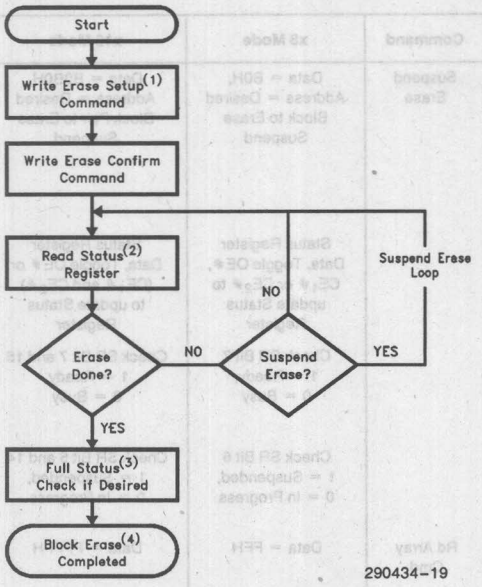
Bus Operation	Command	x8 Mode	x16 Mode
Write	Write Setup	Data = 40H Address = Byte Within Card to be Written	Data = 4040H Address = Word Within Card to be Written
Write	Data Write	Data to be Written Address = Byte Within Card to be Written	Data to be Written Address = Word Within Card to be Written
Read	Defaults to Device Status Register Read Mode	Status Register Data. Toggle OE#, CE_1# or CE_2# to update Status Register	Status Register Data. Toggle OE# or (CE_1# and CE_2#) to update Status Registers
Standby		Check SR Bit 7 1 = Ready, 0 = Busy	Check SR Bits 7 and 15 1 = Ready, 0 = Busy



Bus Operation	Command	x8 Mode	x16 Mode
Standby		Check SR Bit 3 1 = V_{PP} Detected Low	Check SR Bits 3 and 11 1 = V_{PP} Detected Low
Standby		Check SR Bit 4 1 = Data Write Error	Check SR Bits 4 and 12 1 = Data Write Error

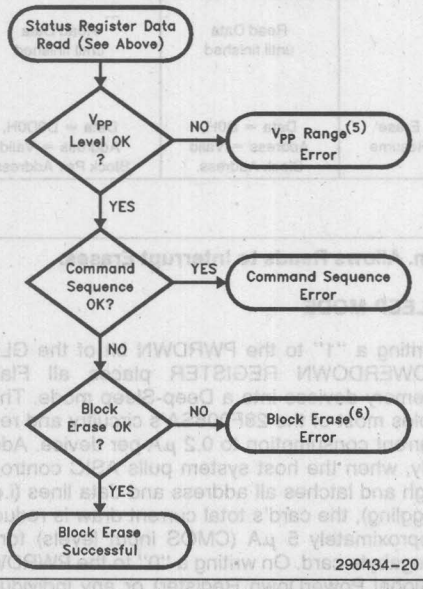
Figure 13. Device-Level Automated Write Algorithm

NOTES:
1. Repeat for subsequent data writes.
2. In addition, the card's READY-BUSY STATUS REGISTER or the RDY/BSY# pin may be used.
3. Full device-level status check can be done after each data write or after a sequence of data writes.
4. Write FFH (or FFFFH) after the last data write operation to reset the device(s) to Read Array Mode.
5. If a data write operation fails due to a low V_{PP} (setting SR Bit 3), the Clear Status Register command MUST be issued before further attempts are allowed by the Write State Machine.
6. If a data write operation fails during a multiple write sequence, SR Bit 4 (Write Status) will not be cleared until the Command User Interface receives the Clear Status Register command.



Bus Operation	Command	x8 Mode	x16 Mode
Write	Erase Setup	Data = 20H Address = Block Within Card to be Erased	Data = 2020H Address = Block Pair Within Card to be Erased
Write	Erase	Data = D0H Address = Block Within Card to be Erased	Data = D0D0H Address = Block Pair Within Card to be Erased
Read	Defaults to Device Status Register Read Mode	Status Register Data. Toggle OE#, CE_1# or CE_2# to update Status Register	Status Register Data. Toggle OE# or (CE_1# and CE_2#) to update Status Register
Standby		Check SR Bit 7 1 = Ready, 0 = Busy	Check SR Bits 7 and 15 1 = Ready, 0 = Busy

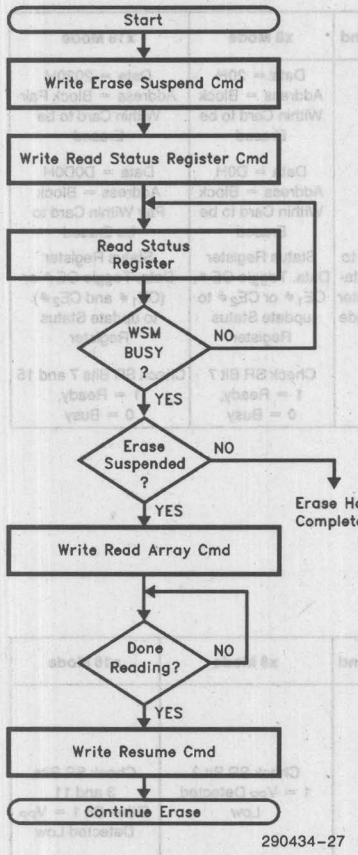
FULL STATUS CHECK PROCEDURE



Bus Operation	Command	x8 Mode	x16 Mode
Standby		Check SR Bit 3 1 = V_{PP} Detected Low	Check SR Bits 3 and 11 Either Bit 1 = V_{PP} Detected Low
Standby		Check SR Bits 4 and 5 Both 1 = Command Sequence Error	Check SR Bits 4, 5, 12, 13 All 1 = Command Sequence Error
Standby		Check SR Bit 5 1 = Block Erase Error	Check SR Bits 5 and 13 Both 1 = Block Erase Failure

Figure 14. Device-Level Automated Erase Algorithm

NOTES:
1. Repeat for subsequent data writes.
2. In addition, the card's READY-BUSY STATUS REGISTER or the RDY/BSY# pin may be used.
3. Full device-level status check can be done after each block erase or after a sequence of block erases.
4. Write FFH (or FFFFH) after the last block erase operation to reset the device(s) to Ready Array Mode.
5. If a block erase operation fails due to a low V_{PP} (setting SR Bit 3), the Clear Status Register command MUST be issued before further attempts are allowed by the Write State Machine.
6. If a block erase operation fails during a multiple block erase sequence, SR Bit 4 (Write Status) will not be cleared until the Command User Interface receives the Clear Status Register command.



Bus Operation	Command	x8 Mode	x16 Mode
Write	Suspend Erase	Data = B0H, Address = Desired Block to Erase Suspend	Data = B0B0H, Address = Desired Block Pair to Erase Suspend
Read		Status Register Data. Toggle OE#, CE_1# or CE_2# to update Status Register	Status Register Data. Toggle OE# or (CE_1# and CE_2#) to update Status Register
Standby		Check SR Bit 7 1 = Ready, 0 = Busy	Check SR Bit 7 and 15 1 = Ready, 0 = Busy
Standby		Check SR Bit 6 1 = Suspended, 0 = In Progress	Check SR Bit 6 and 14 1 = Suspended, 0 = In Progress
Write	Rd Array Cmd	Data = FFH	Data = FFFFH
Read		Read Data until finished	Read Data until finished
Write	Erase Resume	Data = D0H, Address = Valid Block Address.	Data = D0D0H, Address = Valid Block Pair Address.

Figure 15. Erase Suspend/Resume Algorithm. Allows Reads to Interrupt Erases.

POWER CONSUMPTION

STANDBY MODE

In most applications, software will only be accessing one device pair at a time. The Series 2 Card is defined to be in the standby mode when one device pair is in the Read Array Mode while the remaining devices are in the Deep-Sleep Mode. The Series 2 Card's CE_1# and CE_2# input signals must also be at V_{IH}. In standby mode, much of the card's circuitry is shut off, substantially reducing power consumption. Typical power consumption for a 20 Megabyte Series 2 card in standby mode is 65 μA.

SLEEP MODE

Writing a "1" to the PWRDWN bit of the GLOBAL POWERDOWN REGISTER places all FlashFile Memory devices into a Deep-Sleep mode. This disables most of the 28F008SA's circuitry and reduces current consumption to 0.2 μA per device. Additionally, when the host system pulls ASIC control logic high and latches all address and data lines (i.e., not toggling), the card's total current draw is reduced to approximately 5 μA (CMOS input levels) for a 20 Megabyte card. On writing a "0" to the PWRDWN bit (Global PowerDown Register) or any individual device pair (Sleep Control Register), a Deep-Sleep mode recovery period must be allowed for 28F008SA device circuitry to power back on.

SYSTEM DESIGN CONSIDERATIONS

POWER SUPPLY DECOUPLING

Flash memory power-switching characteristics require careful device decoupling. System designers are interested in three supply current issues—standby, active and transient current peaks, produced by rising and falling edges of CE_1# and CE_2#. The capacitive and inductive loads on the card and internal flash memory device pairs determine the magnitudes of these peaks.

The Flash Memory Card features on-card ceramic decoupling capacitors connected between V_{CC} and GND, and between V_{PP1}/V_{PP2} and GND to help transient voltage peaks.

On the host side, the card connector should also have a 4.7 μF electrolytic capacitor between V_{CC} and GND, as well as between V_{PP1}/V_{PP2} and GND. The bulk capacitors will overcome voltage slumps caused by printed-circuit-board trace inductance, and will supply charge to the smaller capacitors as needed.

POWER UP/DOWN PROTECTION

Each device in the Flash Memory Card is designed to offer protection against accidental erasure or writing, caused by spurious system-level signals that may exist during power transitions. The card will power-up into the Read Array Mode.

A system designer must guard against active writes for V_{CC} voltages above V_{LKO} when V_{PP} is active. Since both WE# and CE_1# (and/or CE_2#) must be low for a command write, driving either to V_{IH} will inhibit writes. With its Command User Interface, alteration of device contents only occurs after successful completion of the two-step command sequences.

While these precautions are sufficient for most applications, an alternative approach would allow V_{CC} to reach its steady state value before raising V_{PP1}/V_{PP2} above V_{CC} + 2.0V. In addition, upon powering-down, V_{PP1}/V_{PP2} should be below V_{CC} + 2.0V, before lowering V_{CC}.

HOT INSERTION/REMOVAL

The capability to remove or insert PC cards while the system is powered on (i.e., hot insertion/removal) requires careful design approaches on the system and card levels. To design for this capability consider card overvoltage stress, system power droop and control line stability.

A PCMCIA/JEIDA specified socket properly sequences the power supplies to the flash memory card via shorter and longer pins. This assures that hot insertion and removal will not result in card damage or data loss.

PCMCIA CARD INFORMATION STRUCTURE

The Card Information Structure (CIS) starts at address zero of the card's Attribute Memory Plane. It contains a variable-length chain of data blocks (tuples) that conform to a basic format as shown in Table 5. This section describes each tuple contained within the Series 2 Flash Memory Card.

The Device Information Tuple

This tuple (CISTPL_DEV = 01H) contains information pertaining to the card's speed and size. The Series 2 Card is offered with a 150 nanosecond access time. Card sizes range between 2 and 20 Megabytes.

Table 5. Tuple Format

Bytes	Data
0	Tuple Code: CISTPL_xxx. The tuple code 0FFH indicates no more tuples in the list.
1	Tuple Link: TPL_LINK. Link to the next tuple in the list. This can be viewed as the number of additional bytes in tuple, excluding this byte. If the link field is zero, the tuple body is empty. If the link field contains 0FFH, this tuple is the last tuple in the list.
2−n	Bytes specific to this tuple.

The Device Geometry Tuple

This tuple (CISTPL_DEVICEGEO = 1EH) is conceptually similar to a DOS disk geometry tuple (CISTPL_GEOMETRY), except it is not a format-dependent property; this deals with the fixed architecture of the memory device(s).

Fields are defined as follows:

DGTPL BUS—Value = n, where system bus width = $2^{(n-1)}$ bytes. N = 2 for standard PCMCIA Release 1.0/2.0 cards.

DGTPL EBS—Value = n, where the memory array's physical memory segments have a minimum erase block size of $2^{(n-1)}$ address increments of DGTPL_BUS-wide accesses.

DGTPL RBS—Value = n, where the memory array's physical memory segments have a minimum read block size of $2^{(n-1)}$ address increments of DGTPL_BUS-wide accesses.

DGTPL WBS—Value = n, where the memory array's physical memory segments have a minimum write block size of $2^{(n-1)}$ address increments of DGTPL_BUS-wide accesses.

DGTPL PART—Value = n, where the memory array's physical memory segments can have partitions subdividing the arrays in minimum granularity of $2^{(n-1)}$ number of erase blocks.

FL DEVICE INTERLEAVE—Value = n, where card architectures employ a multiple of $2^{(n-1)}$ times interleaving of the entire memory arrays with the above characteristics. Non-interleaved cards have values n = 1.

Jedec Programming Information Tuple

This tuple (CISTPL_JEDEC = 18H) contains the Intel manufacturing identifier (89H) and the 28F008SA device ID (A2H).

Level 1 Version/Product Information Tuple

This tuple (CISTPL_VERI = 15H) contains Level-1-version compliance and card-manufacturer information. Fields are described as follows:

TPLLV1 MAJOR—Major version number = 04H.

TPLLV1 MINOR—Minor version number = 01H for release 2.0.

TPLLV1 INFO—

Name of manufacturer	= intel;
Name of product	= SERIES2-"Card size";
Card type	= 2;
Speed	= 150 ns or 200 ns
Register Base	= REGBASE 4000H
Test Codes	= DBBDRELP
Legalities	= COPYRIGHT intel Corporation 1991

The Configurable Card Tuple

This tuple (CISTPL_CONF = 1AH) describes the interface supported by the card and the locations of the Card Configuration Registers and the Card Configuration Table.

Fields are described as follows:

TPCC SZ—Size of fields byte = 01H.

TPCC LAST—Index number of the last entry in the Card Configuration Table = 00H.

TPCC RADR—Configuration Registers Base Address in Reg Space = 4000H.

TPCC RMSK—Configuration Registers Present Mask = 03H.

The End-Of-List Tuple

The end-of-list tuple (CISTPL_END = FFH) marks the end of a tuple chain. Upon encountering this tuple, continue tuple processing as if a long-link to address 0 of common memory space were encountered.

Tuple Address	Value	Description
00H	01H	CISTPL_DEV
02H	03H	TPL_LINK
04H	53H	DEVICE_INFO = FLASH 150 ns
	52H	DEVICE_INFO = FLASH 200 ns
06H		CARD SIZE
	06H	2M
	0EH	4M
	26H	10M
	4EH	20M
08H	FFH	END OF DEVICE
0AH	1EH	CISTPL_ DEVICEGEO
0CH	06H	TPL_LINK
0EH	02H	DGTPL_BUS
10H	11H	DGTPL_EBS
12H	01H	DGTPL_RBS
14H	01H	DGPL_WBS
16H	03H	DGTPL_PART
18H	01H	FL_DEVICE INTERLEAVE
1AH	18H	CISTPL_JEDEC
1CH	02H	TPL_LINK
1EH	89H	INTEL J-ID
20H	A2H	28F008 J-ID
22H	15H	CISTPL_VER1
24H	50H	TPL_LINK
26H	04H	TPLLV1 MAJOR
28H	01H	TPLLV1 MINOR
2AH	69H	TPLLV1 INFO i
2CH	6EH	n
2EH	74H	t
30H	65H	e

Tuple Address	Value	Description
32H	6CH	l
34H	00H	END TEXT
36H	53H	S
38H	45H	E
3AH	52H	R
3CH	49H	I
3EH	45H	E
40H	53H	S
42H	32H	2
44H	2DH	—
46H	30H	2M = 0
	30H	4M = 0
	31H	10M = 1
	32H	20M = 2
48H	32H	2M = 2
	34H	4M = 4
	30H	10M = 0
	30H	20M = 0
4AH	20H	SPACE
4CH	00H	END TEXT
4EH	32H	CARD TYPE 2
50H	41H	A = 2M, 150 ns
	42H	B = 4M, 150 ns
	45H	E = 10M, 150 ns
	5AH	Z = 20M, 150 ns
	48H	H = 2M, 200 ns
	49H	I = 4M, 200 ns
	4CH	L = 10M, 200 ns
	4FH	O = 20M, 200 ns

Tuple Address	Value	Description
52H	20H	SPACE
54H	52H	REGBASE-R
56H	45H	E
58H	47H	G
5AH	42H	B
5CH	41H	A
5EH	53H	S
60H	45H	E
62H	20H	SPACE
64H	34H	4000h 4
66H	30H	0
68H	30H	0
6AH	30H	0
6CH	68H	h
6EH	20H	SPACE
70H	44H	D
72H	42H	B
74H	42H	B
76H	44H	D
78H	52H	R
7AH	45H	E
7CH	4CH	L
7EH	50H	P
80H	00H	END TEXT
82H	43H	COPYRIGHT C
84H	4FH	O
86H	50H	P
88H	59H	Y
8AH	52H	R
8CH	49H	I
8EH	47H	G
90H	48H	H
92H	54H	T
94H	20H	SPACE

Tuple Address	Value	Description
96H	69H	i
98H	6EH	n
9AH	74H	t
9CH	65H	e
9EH	6CH	l
A_0H	20H	SPACE
A2H	43H	CORPORATION C
A4H	4FH	O
A6H	52H	R
A8H	50H	P
AAH	4FH	O
ACH	52H	R
AEH	41H	A
B0H	54H	T
B2H	49H	I
B4H	4FH	O
B6H	4EH	N
B8H	20H	SPACE
BAH	31H	1
BCH	39H	9
BEH	39H	9
C0H	31H	1
C2H	00H	END TEXT
C4H	FFH	END OF LIST
C6H	1AH	CISTPL__CONF
C8H	06H	TPL__LINK
CAH	01H	TPCC__SZ
CCH	00H	TPCC__LAST
CEH	00H	TPCC__RADR
D0H	40H	TPCC__RADR
D2H	03H	TPCC__RMSK
D4H	FFH	END OF LIST
D6H	FFH	CISTPL__END
D8H	00H	INVALID ECIS ADDRESS

OPERATING SPECIFICATIONS

NOTICE: This data sheet contains preliminary information on new products in production. The specifications are subject to change without notice. Verify with your local Intel Sales office that you have the latest data sheet before finalizing a design.

ABSOLUTE MAXIMUM RATINGS*

Storage Temperature . . . . . −40°C to +85°C

Voltage on Any Pin with Respect to Ground . . . . . −2.0V to V_{CC} + 2.0V(1)

V_{PP1}/V_{PP2} Supply Voltage with Respect to Ground . . . . . −2.0V to +14.0V(1, 2)

V_{CC} Supply Voltage with Respect to Ground . . . . . −0.5V to +7.0V

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

NOTES:
1. Minimum DC input voltage is −0.5V. During transitions, inputs may undershoot to −2.0V tor periods of less than 20 ns. Maximum DC voltage on output pins is V_{CC} + 0.5V. The voltage may overshoot to V_{CC} + 2.0V for periods of less than 20 ns.
2. Maximum DC input voltage on V_{PP1}/V_{PP2} may overshoot to +14.0V for periods of less than 20 ns.

COMMERCIAL TEMPERATURE OPERATING CONDITIONS

These operating conditions apply to commercial temperature devices.

Symbol	Parameter	Min	Max	Unit
T_A	Operating Temperature	0	70	°C
V_{CC}	V_{CC} Supply Voltage (5%)	4.75	5.25	V

EXTENDED TEMPERATURE OPERATING CONDITIONS

These operating conditions apply to extended temperature devices.

Symbol	Parameter	Min	Max	Unit
T_A	Operating Temperature	−40	85	°C
V_{CC}	V_{CC} Supply Voltage (5%)	4.75	5.25	V

CHARACTERISTICS

All AC and DC characteristics apply to both commercial and extended temperature devices.

COMMON DC CHARACTERISTICS, CMOS and TTL

Symbol	Parameter	Notes	Min	Typ	Max	Unit	Test Condition
I_{LI}	Input Leakage Current	1, 3		±1	±20	µA	$V_{CC} = V_{CC}$ Max $V_{IN} = V_{CC}$ or GND
I_{LO}	Output Leakage Current	1		±1	±20	µA	$V_{CC} = V_{CC}$ Max $V_{OUT} = V_{CC}$ or GND
V_{IL}	Input Low Voltage	1	−0.5		0.8	V	
V_{IH}	Input High Voltage (TTL)	1	2.4		V_{CC} + 0.3	V	
	Input High Voltage (CMOS)		0.7 V_{CC}		V_{CC} + 0.3		
V_{OL}	Output Low Voltage	1	V_{SS}		0.4	V	$V_{CC} = V_{CC}$ Min I_{OL} = 3.2 mA
V_{OH}	Output High Voltage	1	4.0		V_{CC}	V	$V_{CC} = V_{CC}$ Min I_{OH} = −2.0 mA
V_{PPL}	V_{PP} during Read Only Operations	1, 2	0.0		6.5	V	
V_{PPH}	V_{PP} during Read/Write Operations	1	11.4		12.6	V	
V_{LKO}	V_{CC} Erase/Write Lock Voltage	1	2.0			V	

NOTES:
1. Values are the same for byte and word wide modes and for all card densities.
2. Block Erases/Data Writes are inhibited when $V_{PP} = V_{PPL}$ and not guaranteed in the range between V_{PPH} and V_{PPL}.
3. Exceptions: With V_{IN} = GND, the leakage on CE_1#, CE_2#, REG#, OE#, WE#, will be ≤ 500 µA due to internal pullup resistors and, with $V_{IN} = V_{CC}$, RST leakage will be ≤ 500 µA due to internal pulldown resistor.

DC CHARACTERISTICS, CMOS

Symbol	Parameter		Notes	Byte Wide Mode			Word Wide Mode			Unit	Test Condition
				Min	Typ	Max	Min	Typ	Max		
I_{CCR}	V_{CC} Read Current		1, 3		45	85		65	120	mA	$V_{CC} = V_{CC}$ Max, Control Signals = GND t_{CYCLE} = 200 ns, I_{OUT} = 0 mA
I_{CCW}	V_{CC} Write Current		1, 3		35	80		45	110	mA	Data Write in Progress
I_{CCE}	V_{CC} Erase Current		1, 2, 3		35	80		45	110	mA	Block (Pair) Erase in Progress
I_{CCS}	V_{CC} Standby Current	2 Meg	1, 4, 6		110	420		110	420	µA	$V_{CC} = V_{CC}$ Max, Control Signals = V_{IH}
		4 Meg			160	620		160	620		
		10 Meg			310	1220		310	1220		
		20 Meg			560	2200		560	2200		
I_{CCSL}	V_{CC} Sleep Current	2 Meg	1, 4, 5			60			60	µA	
		4 Meg				100			100		
		10 Meg				260			260		
		20 Meg				460			460		
I_{PPW}	V_{PP} Write Current ($V_{PP} = V_{PPH}$)		1, 3			15			30	mA	Data Write in Progress
I_{PPE}	V_{PP} Erase Current ($V_{PP} = V_{PPH}$)		1, 3			15			30	mA	Block (Pair) Erase in Progress
I_{PPSL}	V_{PP} Sleep Current	2 Meg	1, 5		0.2	10		0.2	10	µA	
		4 Meg			0.4	20		0.4	20		
		10 Meg			1	50		1	50		
		20 Meg			2	100		2	100		
I_{PPS1}	V_{PP} Standby or Read Current ($V_{PP} \leq V_{CC}$)	2 Meg	1, 6		2.0	20		2.0	20	µA	
		4 Meg			2.2	30		2.2	30		
		10 Meg			2.8	60		2.8	60		
		20 Meg			3.8	110		3.8	110		
I_{PPS2}	V_{PP} Standby or Read Current ($V_{PP} > V_{CC}$)	2 Meg	1, 6		20	400		20	400	µA	
		4 Meg			40	800		40	800		
		10 Meg			100	2000		100	2000		
		20 Meg			200	4000		200	4000		

NOTES:
1. All currents are in RMS unless otherwise noted. Typical values at V_{CC} = 5.0V, V_{PP} = 12.0V, T = 25°C.
2. The Data Sheet specification for the 28F008SA in Erase Suspend (I_{CCES}) is 5 mA typical and 10 mA max with the device deselected. If the device(s) are read while in Erase Suspend Mode, current draw is the sum of I_{CCES} and I_{CCR}.
3. Standby or Sleep currents are not included for non-accessed devices.
4. Address and data inputs to card static. Control line voltages equal to V_{IH} or V_{IL}.
5. All 28F008SA devices in Deep-Sleep (Reset-PowerDown) mode.
6. In Byte and Word Mode, all but two devices in Deep-Sleep.

DC CHARACTERISTICS, TTL

Symbol	Parameter		Notes	Byte Wide Mode			Word Wide Mode			Unit	Test Condition
				Min	Typ	Max	Min	Typ	Max		
I_{CCR}	V_{CC} Read Current		1, 3		70	135		90	170	mA	$V_{CC} = V_{CC}$ Max, Control Signals = GND t_{CYCLE} = 200 ns, I_{OUT} = 0 mA
I_{CCW}	V_{CC} Write Current		1, 3		60	130		70	160	mA	Data Write in Progress
I_{CCE}	V_{CC} Erase Current		1, 2, 3		60	130		70	160	mA	Block (Pair) Erase in Progress
I_{CCS}	V_{CC} Standby Current	2 Meg	1, 4, 6		20	100		20	100	mA	$V_{CC} = V_{CC}$ Max, Control Signals = V_{IH}
		4 Meg									
		10 Meg									
		20 Meg									
I_{CCSL}	V_{CC} Sleep Current	2 Meg	1, 4, 5		20	100		20	100	mA	
		4 Meg									
		10 Meg									
		20 Meg									
I_{PPW}	V_{PP} Write Current ($V_{PP} = V_{PPH}$)		1, 3		10	30		20	60	mA	Data Write in Progress
I_{PPE}	V_{PP} Erase Current ($V_{PP} = V_{PPH}$)		1, 3		10	30		20	60	mA	Block (Pair) Erase in Progress
I_{PPSL}	V_{PP} Sleep Current	2 Meg	1, 5		0.2	10		0.2	10	µA	
		4 Meg			0.4	20		0.4	20		
		10 Meg			1.0	50		1.0	50		
		20 Meg			2.0	100		2.0	100		
I_{PPS1}	V_{PP} Standby or Read Current ($V_{PP} \leq V_{CC}$)	2 Meg	1, 6		2.0	20		2.0	20	µA	
		4 Meg			2.2	30		2.2	30		
		10 Meg			2.8	60		2.8	60		
		20 Meg			3.8	110		3.8	110		
I_{PPS2}	V_{PP} Standby or Read Current ($V_{PP} > V_{CC}$)	2 Meg	1, 6		20	400		20	400	µA	
		4 Meg			40	800		40	800		
		10 Meg			100	2000		100	2000		
		20 Meg			200	4000		200	4000		

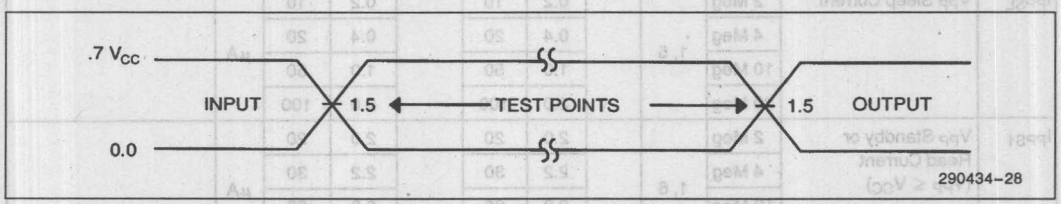
NOTES:
1. All currents are in RMS unless otherwise noted. Typical values at V_{CC} = 5.0V, V_{PP} = 12.0V, T = 25°C.
2. The Data Sheet specification for the 28F008SA in Erase Suspend (I_{CCES}) is 5 mA typical and 10 mA max with the device deselected. If the device(s) are read while in Erase Suspend Mode, current draw is the sum of I_{CCES} and I_{CCR}.
3. Standby or Sleep currents are not included for non-accessed devices.
4. Address and data inputs to card static. Control line voltages equal to V_{IH} or V_{IL}.
5. All 28F008SA devices in Deep-Sleep (Reset-PowerDown) mode.
6. In Byte and Word Mode, all but two devices in Deep-Sleep.
7. The current consumption from the 28F008SA is insignificant in relation to the ASIC's.

AC CHARACTERISTICS

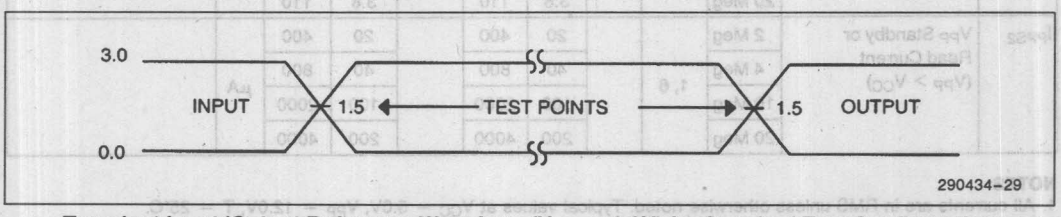
AC Timing Diagrams and characteristics are guaranteed to meet or exceed PCMCIA Release 2.0 specifications. PCMCIA allows a 300 ns access time for Attribute Memory. Note that read and write access timings to the Series 2 Flash Memory Card's Common and Attribute Memory Planes are identical at 150 ns. Furthermore, there is no delay in switching between the Common and Attribute Memory Planes.

COMMON AND ATTRIBUTE MEMORY, AC CHARACTERISTICS: Read-Only Operations

Symbol		Parameter	Notes	Min	Max	Unit
JEDEC	PCMCIA					
t_{AVAV}	t_{RC}	Read Cycle Time		150		ns
t_{AVQV}	t_a (A)	Address Access Time			150	ns
t_{ELQV}	t_a (CE)	Card Enable Access Time			150	ns
t_{GLQV}	t_a (OE)	Output Enable Access Time			75	ns
t_{EHQX}	t_{dis} (CE)	Output Disable Time from CE#			75	ns
t_{GHQZ}	t_{dis} (CE)	Output Disable Time from OE#			75	ns
t_{GLQX}	t_{en} (CE)	Output Enable Time from CE#		5		ns
t_{ELQX}	t_{en} (OE)	Output Enable Time from OE#		5		ns
t_{AXQX}	t_v (A)	Data Valid from Add Change		0		ns
t_{RHQV}		Reset-PwrDwn Recovery to Output Delay		500		ns
	t_{su} (V_{CC})	CE Setup Time on Power-Up		1		ms
		First Access after Reset		500		ns



Transient Input/Output Reference Waveform (V_{CC} = 5.0V) for Standard Test Configuration



Transient Input/Output Reference Waveform (V_{CC} = 3.3V) for Standard Test Configuration

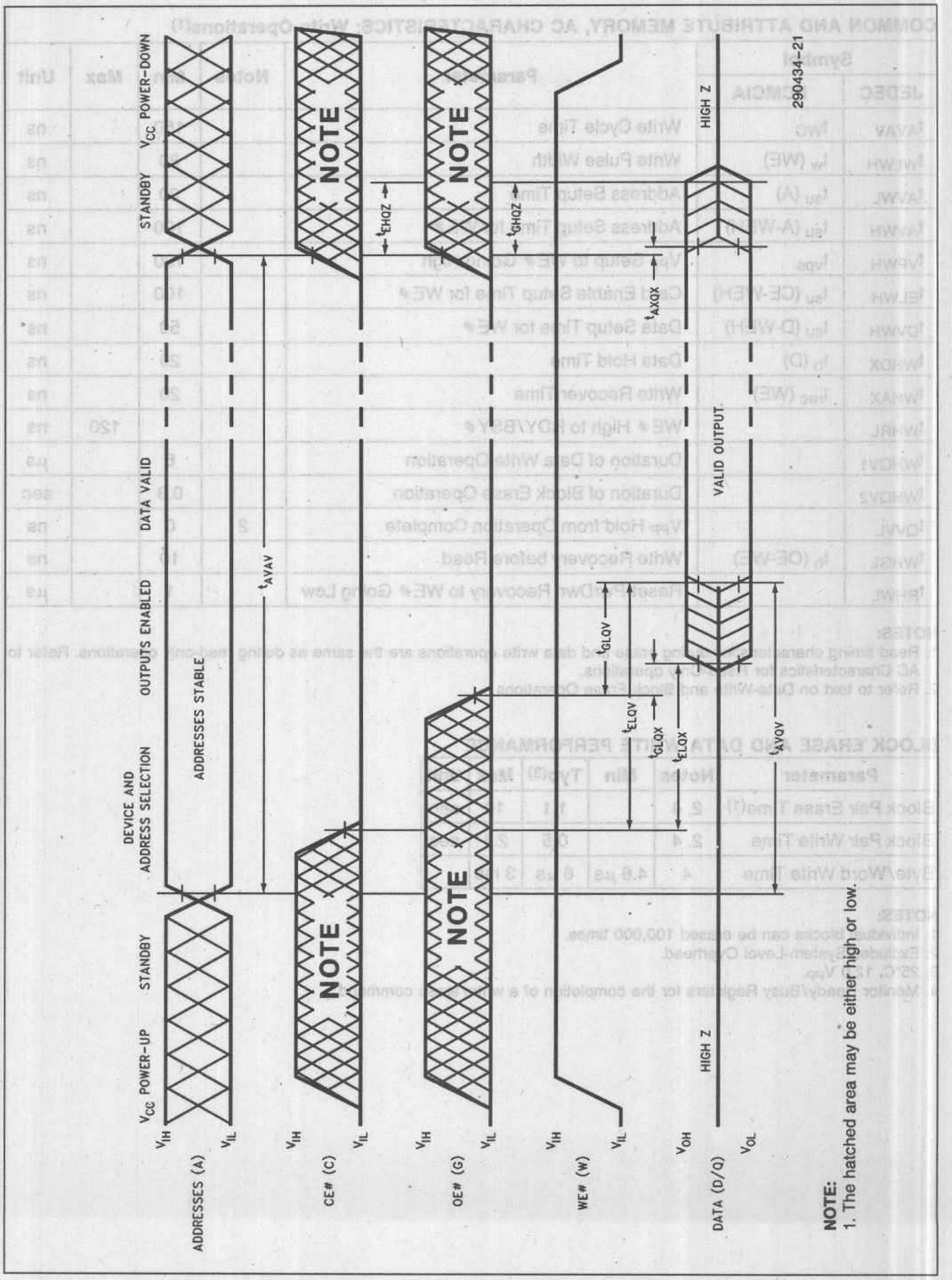


Figure 16. AC Waveform for Read Operations

COMMON AND ATTRIBUTE MEMORY, AC CHARACTERISTICS: Write Operations(1)

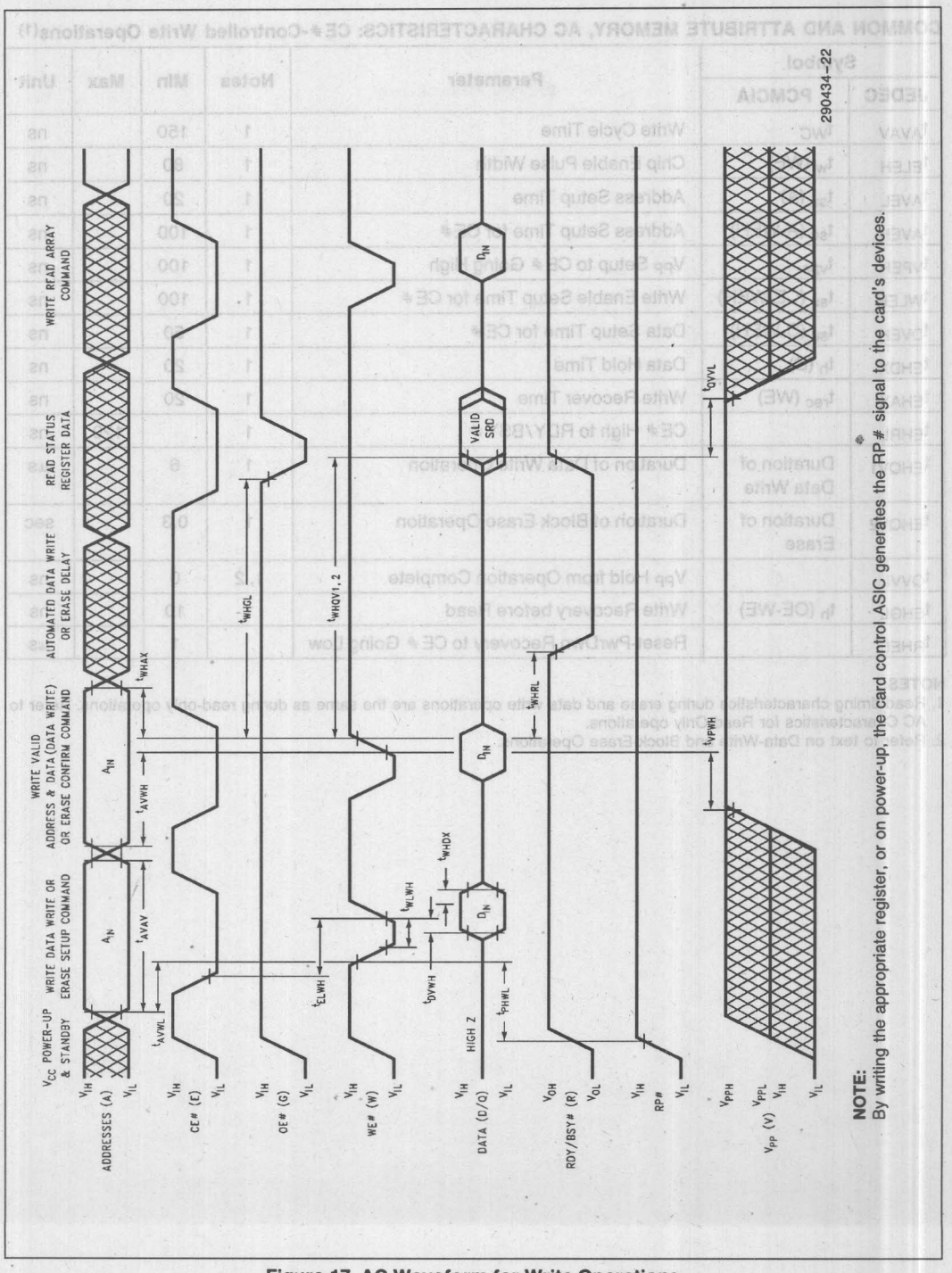
Symbol		Parameter	Notes	Min	Max	Unit
JEDEC	PCMCIA					
t_{AVAV}	t_{WC}	Write Cycle Time		150		ns
t_{WLWH}	t_w (WE)	Write Pulse Width		80		ns
t_{AVWL}	t_{su} (A)	Address Setup Time		20		ns
t_{AVWH}	t_{su} (A-WEH)	Address Setup Time for WE#		100		ns
t_{VPWH}	t_{vps}	V_{PP} Setup to WE# Going High		100		ns
t_{ELWH}	t_{su} (CE-WEH)	Card Enable Setup Time for WE#		100		ns
t_{DVWH}	t_{su} (D-WEH)	Data Setup Time for WE#		50		ns
t_{WHDX}	t_h (D)	Data Hold Time		20		ns
t_{WHAX}	t_{rec} (WE)	Write Recover Time		20		ns
t_{WHRL}		WE# High to RDY/BSY#			120	ns
t_{WHQV1}		Duration of Data Write Operation		6		µs
t_{WHQV2}		Duration of Block Erase Operation		0.3		sec
t_{QVVL}		V_{PP} Hold from Operation Complete	2	0		ns
t_{WHGL}	t_h (OE-WE)	Write Recovery before Read		10		ns
t_{RHWL}		Reset-PwrDwn Recovery to WE# Going Low		1		µs

NOTES:
1. Read timing characteristics during erase and data write operations are the same as during read-only operations. Refer to AC Characteristics for Read-Only operations.
2. Refer to text on Data-Write and Block-Erase Operations.

BLOCK ERASE AND DATA WRITE PERFORMANCE

Parameter	Notes	Min	Typ(3)	Max	Unit
Block Pair Erase Time(1)	2, 4		1.1	10	sec
Block Pair Write Time	2, 4		0.5	2.1	sec
Byte/Word Write Time	4	4.8 µs	6 µs	3 ms	

NOTES:
1. Individual blocks can be erased 100,000 times.
2. Excludes System-Level Overhead.
3. 25°C, 12.0 V_{PP}.
4. Monitor Ready/Busy Registers for the completion of a write/erase command.



NOTE:
By writing the appropriate register, or on power-up, the card control ASIC generates the RP# signal to the card's devices.

Figure 17. AC Waveform for Write Operations

COMMON AND ATTRIBUTE MEMORY, AC CHARACTERISTICS: CE#-Controlled Write Operations(1)

Symbol		Parameter	Notes	Min	Max	Unit
JEDEC	PCMCIA					
t_{AVAV}	t_{WC}	Write Cycle Time	1	150		ns
t_{ELEH}	t_w (WE)	Chip Enable Pulse Width	1	80		ns
t_{AVEL}	t_{su} (A)	Address Setup Time	1	20		ns
t_{AVEH}	t_{su} (A-WEH)	Address Setup Time for CE#	1	100		ns
t_{VPEH}	t_{vps}	V_{PP} Setup to CE# Going High	1	100		ns
t_{WLEH}	t_{su} (CE-WEH)	Write Enable Setup Time for CE#	1	100		ns
t_{DVEH}	t_{su} (D-WEH)	Data Setup Time for CE#	1	50		ns
t_{EHDX}	t_h (D)	Data Hold Time	1	20		ns
t_{EHAX}	t_{rec} (WE)	Write Recover Time	1	20		ns
t_{EHRL}		CE# High to RDY/BSY#	1		120	ns
t_{EHQV1}	Duration of Data Write	Duration of Data Write Operation	1	6		µs
t_{EHQV2}	Duration of Erase	Duration of Block Erase Operation	1	0.3		sec
t_{QVVL}		V_{PP} Hold from Operation Complete	1, 2	0		ns
t_{EHGL}	t_h (OE-WE)	Write Recovery before Read	1	10		ns
t_{RHEL}		Reset-PwrDwn Recovery to CE# Going Low		1		µs

NOTES:
1. Read timing characteristics during erase and data write operations are the same as during read-only operations. Refer to AC Characteristics for Read-Only operations.
2. Refer to text on Data-Write and Block-Erase Operations.

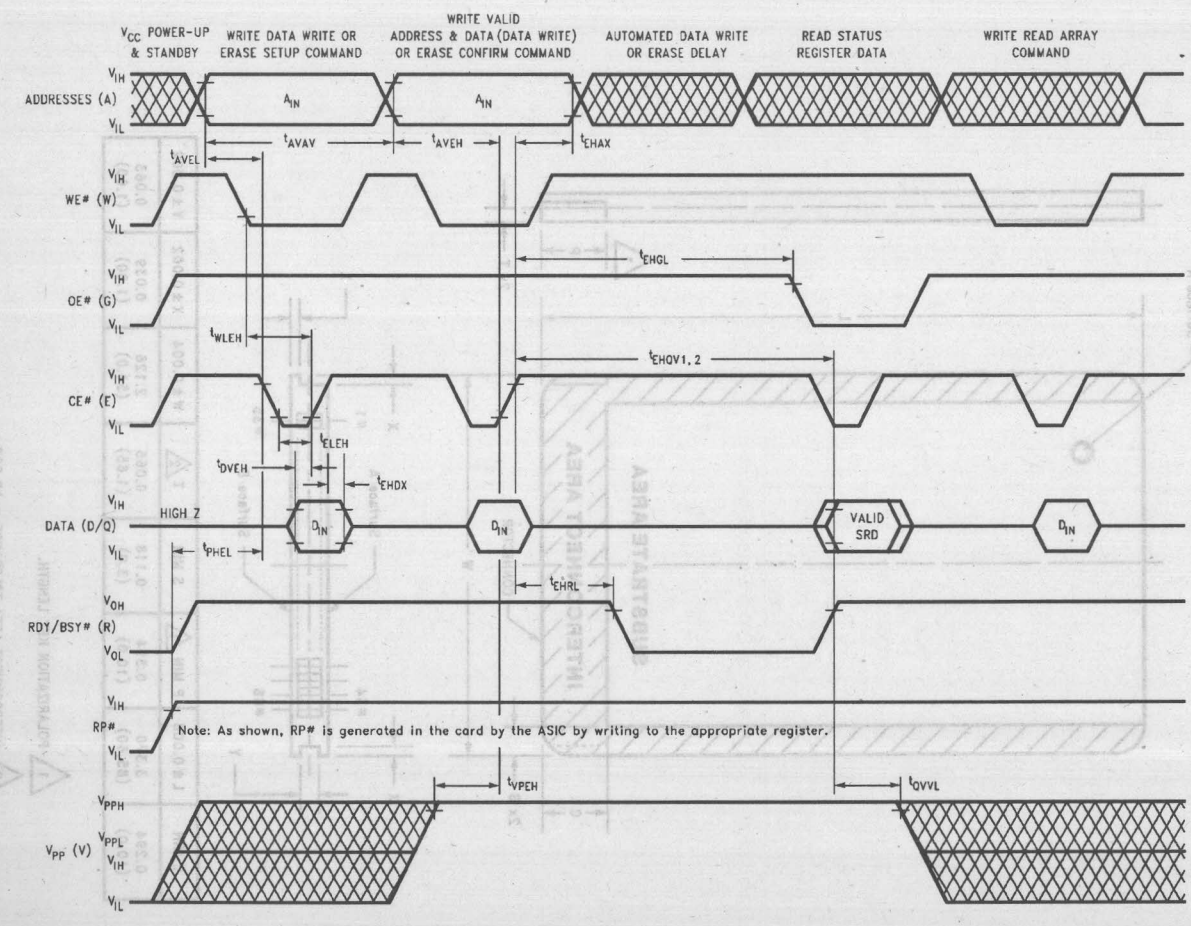
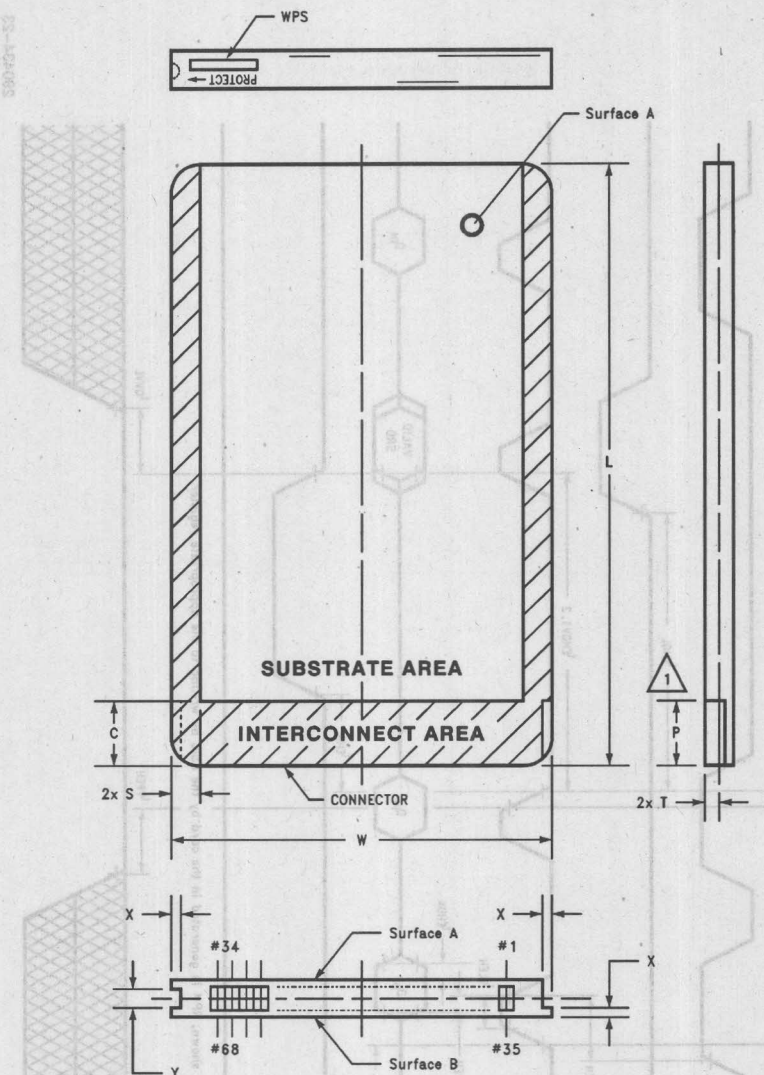


Figure 18. Alternate AC Waveform for Write Operations

NOTE:
By writing the appropriate register, or on power-up, the card control ASIC generates the RP# signal to the card's devices.



C MIN	L ± 0.008	P MIN 1	S MIN	T 2	W ± 0.004	X ± 0.002	Y ± 0.002
0.294 (10.0)	3.370 (85.60)	0.394 (10.0)	0.118 (3.0)	0.065 (1.65)	2.126 (54.0)	0.039 (1.00)	0.063 (1.60)

1. POLARIZATION KEY LENGTH.
2. INTERCONNECT AREA TOLERANCE = ±0.002
   SUBSTRATE AREA TOLERANCE = ±0.004
3. MILLIMETERS ARE IN PARENTHESIS ().

290434-24

Figure 19. Series 2 Flash Memory Card Package Dimensions

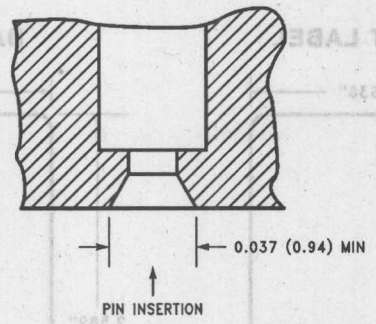
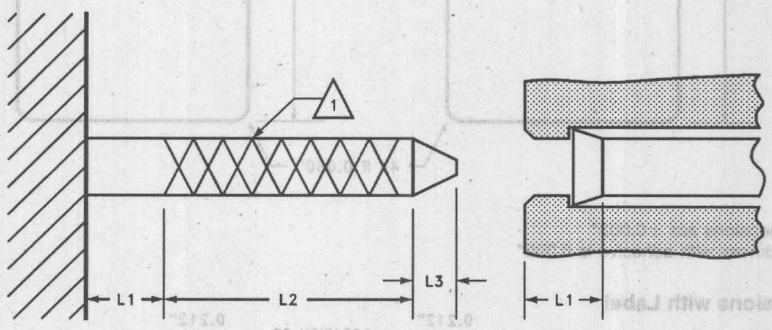


Figure 20. Card Connector Socket



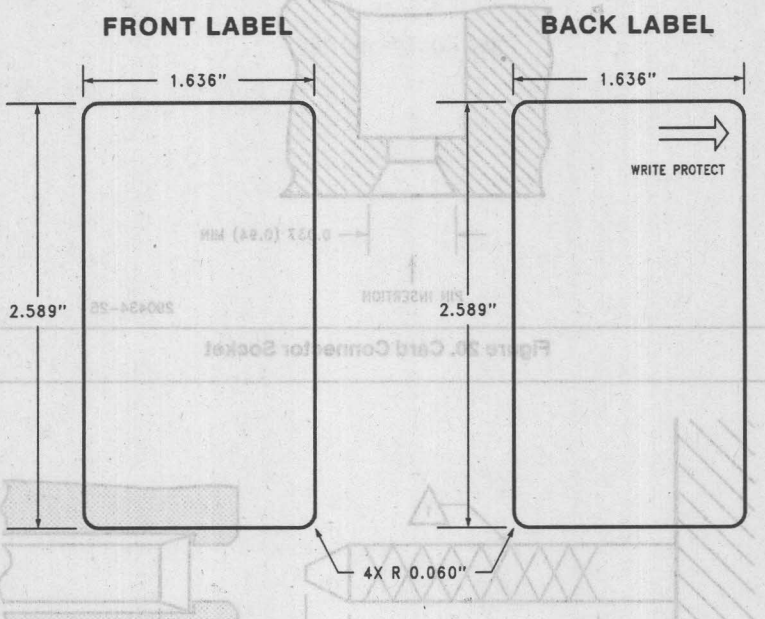
1. PIN/SOCKET CONTACT AREA
2. MILLIMETERS ARE IN PARENTHESIS ()

290434-26

L1 MAX	L2	L3 REF
0.020 (0.5)	Pin Type—See Table 1 Detect 0.059 (1.5) ±0.039 General 0.084 (2.1) ±0.064 Power 0.098 (2.5) ±0.078	0.024 (0.6)

Figure 21. Pin/Socket Contact Length with Wipe

Label Dimensions



NOTES:
Total label dimensions are ±0.003"
Total label thickness with adhesive is 0.002"

Card Dimensions with Label

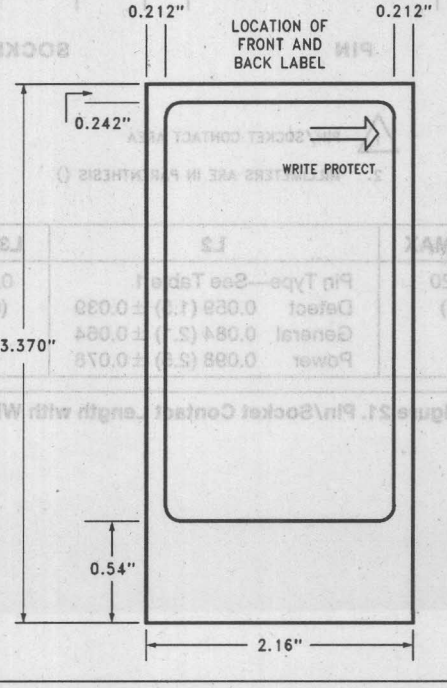


Figure 22. Label Information

Table 5. Capacitance $T_A = 25°C$, f = 1.0 MHz

Symbol	Characteristics	Commercial		Unit
		Min	Max	
C_{IN}	Address/Control Capacitance (A_0–A_8, CE_1#, CE_2#)		30	pF
	Address/Control Capacitance (A_9–A_{24}, all others)		20	pF
	V_{CC}, V_{PP}		2	μF
C_{OUT}	Output Capacitance		20	pF

NOTE:
Sampled, not 100% tested.

ORDERING INFORMATION

iMC020FLSA-15 M9508014,SBXXXX

WHERE:

i	= INTEL	M	= MANILA
MC	= MEMORY CARD	95	= 1995
020	= DENSITY IN MEGABYTES (002,004,010,020 AVAILABLE)	08	= WEEK 08
		014	= LOT #
FL	= FLASH TECHNOLOGY	SBXXXX	= CUSTOMER IDENTIFIER
S	= BLOCKED ARCHITECTURE		
A	= SERIES 2		
-ET	= EXTENDED TEMPERATURE		
15	= 150 ns		

ADDITIONAL INFORMATION	ORDER NUMBER
28F008SA FlashFile™ Memory Data Sheet	290429
AP-361 "Implementing the Integrated Registers of the Series 2 Flash Memory Card"	292096
AP-364 "28F008SA Automation and Algorithms"	292099
AP-359 "28F008SA Hardware Interfacing"	292094
AP-360 "28F008SA Software Drivers"	292095
AP-606 "Interchangeability of Series 1/2/2+ Flash Memory Cards"	

REVISION HISTORY

Number	Description
02	Added 150 ns TUPLE, Deleted 250 ns TUPLE Corrected Global Power Register Address to 4002H Corrected Write Protection Register Address to 4104H Corrected Ready-Busy Mode Register Address to 4140H I_{CC} Standby Byte Wide Mode MAX/TYP Increased Added Power-On Timing Spec Added First Access after Reset Spec Changed Advanced Information to Preliminary
03	Added 2 MByte card support Changed write timing waveforms to match PCMCIA Changed PowerDown (PWD) to Reset-PowerDown (RP)
04	Extended Operating Temperature Range Added Maximum Byte Write Time
05	Added 150 ns Timings Change Cover Drawing to Accommodate Label
06	Modified DC characteristics to reflect a conversion to new memory components—the 0.4μ version of the 28F008SA. The new devices also have improved program and erase performance.

Table 5. Capacitance T_A = 25°C, f = 1.0 MHz

Symbol	Characteristics	Commercial		Unit
		Min	Max	
C_{IN}	Address/Control Capacitance (A_0–A_{25}, $\overline{CE_1}$, $\overline{CE_2}$)		30	pF
	Address/Control Capacitance (A_9–A_{25}, all others)		20	pF
	V_{CC}, V_{PP}		2	µF
C_{OUT}	Output Capacitance		20	pF

NOTE:
Sampled, not 100% tested.

ORDERING INFORMATION

M28208014,SBXXXX

IM-28F020FLSA-15

WHERE:

I	= INTEL	M	= MANILA
MC	= MEMORY CARD	95	= 1995
020	= DENSITY IN MEGABYTES (002,004,010,020 AVAILABLE)	08	= WEEK 08
		014	= LOT #
FL	= FLASH TECHNOLOGY	SBXXXX	= CUSTOMER IDENTIFIER
S	= BLOCKED ARCHITECTURE		
A	= SERIES 2		
-ET	= EXTENDED TEMPERATURE		
15	= 150 ns		

ADDITIONAL INFORMATION

	ORDER NUMBER
28F008SA FlashFile™ Memory Data Sheet	290429
AP-361 "Implementing the Integrated Registers of the Series 2 Flash Memory Card"	292096
AP-364 "28F008SA Automation and Algorithms"	292095
AP-365 "28F008SA Hardware Interfacing"	292094
AP-360 "28F008SA Software Drivers"	292098
AP-606 "Interchangeability of Series 1/2/2+ Flash Memory Cards"	

REVISION HISTORY

Number	Description
02	Added 150 ns TURPLE, Deleted 250 ns TURPLE Corrected Global Power Register Address to 4002H Corrected Write Protection Register Address to 4104H Corrected Ready-Busy Mode Register Address to 4140H I_{CC} Standby Byte Wide Mode MAX/TYP Increased Added Power-On Timing Spec Added First Access after Reset Spec Changed Advanced Information to Preliminary
03	Added 2 MByte card support Changed write timing waveforms to match PCMCIA Changed PowerDown (PWD) to Reset-PowerDown (RP)
04	Extended Operating Temperature Range Added Maximum Byte Write Time
05	Added 150 ns Timings Change Cover Drawing to Accommodate Label
06	Modified DC characteristics to reflect a conversion to new memory components—the 0.4µ version of the 28F008SA. The new devices also have improved program and erase performance.

PRELIMINARY

intel®

3

Development Support Tools

intel.

3

Development Support Tools

i960® Microprocessor CTOOLS Application Development Tools

PRODUCT OVERVIEW

Intel's CTOOLS provides a complete set of application development tools for developing embedded designs, including an advanced optimizing compiler, an assembler, a linker, utilities, and a variety of libraries, including floating-point emulation.

Besides operating with the most popular operating systems, CTOOLS also incorporates industry standards in all areas. CTOOLS conforms to ANSI Standard X3.159-1989 and passes the Plum Hall conformance and Perennial test suites. CTOOLS also conforms to the 80960 Application Binary Interface (ABI), enabling object code interoperability with third-party tools and debuggers. Compatibility with known standards makes new users productive immediately, and ensures access to existing application code.

CTOOLS can be used across all members of the i960® microprocessor family. Command line switches allow the compiler to take advantage of specific architectural features. For instance, in the case of the i960 Cx and Hx processors, the compiler uses advanced code scheduling algorithms to modify instruction sequences, taking advantage of the processor's parallel execution capability. The generated code is highly efficient, assuring maximum performance for your embedded applications.

PROVEN OPTIMIZATION TECHNIQUES

Advanced optimization techniques are incorporated into Intel's CTOOLS compiler to offer customers superior performance while maintaining robust code. The compiler incorporates local, global, program-wide and profile-driven optimizations:

Processor-Independent Optimizations Including:

- Constant expression evaluation
- Constant propagation
- Collapsing of arithmetic and bitwise boolean identities
- Common subexpression elimination
- Register subsumption or register coalescing
- Local variable promotions
- Tail-call elimination
- Procedure inlining
- Branch optimizations
- Dead code elimination
- Loop invariant code motion
- Variable shadowing
- Superblock formation
- Basic block rearrangement

Processor-Dependent Optimizations Including:

- Specialized instruction selection
- An intelligent register manager
- Code scheduling
- Use of on-chip data RAM for spill registers
- Efficient use of complex addressing modes
- Branch prediction
- Generation of leaf procedures
- Memory access coalescing

WHOLE-PROGRAM AND PROFILE-DRIVEN OPTIMIZATIONS

CTOOLS also provides program-level optimizations, which allow optimizations such as function inlining to occur across source files. Changing the optimization level is as simple as changing an argument on the compiler's command line. Restructuring your build environment is no longer needed!

A runtime profile can be used to guide the whole-program optimization decisions. Such profile-driven optimizations combine a global view of the entire program with its typical runtime behavior, to produce highly optimized code.

Collecting a runtime profile is often an expensive procedure. With CTOOLS R5.0, once a runtime profile is collected, it can be used to guide optimizations after days, weeks or even months of changes to your source code. The profile is automatically interpolated to match the structure of your program.

COMPRESSION ASSISTED VIRTUAL EXECUTION (CAVE)

By storing non-critical functions in compressed form, CTOOLS can save valuable memory in your ROM-based application. When invoked, such functions are decompressed onto the runtime stack and executed. Upon function return, the stack space is automatically freed.

DEBUGGING OPTIMIZED CODE

The DWARF 2.0 symbolic debug information format supports expression of the complex relationships between your source program and its highly optimized object code. Debugging features such as breakpoints and displaying a variable's value behave more reliably with DWARF than with existing object file formats such as COFF and b.out, when your code is optimized. Optimized code debugging can often eliminate the expensive step of building an unoptimized version of your application for debugging purposes.

KEY FEATURES

- Improved Code Generation for the i960® RP, Jx and Hx Processor Families
- Easy-to-Use Whole-Program and Profile-Driven Optimizations
- Efficient Memory Use with Runtime Decompression of Compressed Object Code
- Debug of Optimized Code Using the ELF Object File Format with DWARF 2.0 Symbolic Debug Records
- Conformance to the 80960 Tools Consortium's Application Binary Interface (ABI) Enhances Interoperability
- PCI Download and PCI Comm on DOS and Windows 95
- On-Line HTML Hypertext Documentation
- Compatible with GNU/960 R4.6 and CTOOLS960 R4.6
- Conforms to ANSI Standard X3.159-1989 and Passes Plum Hall Conformance and Perennial Tests
- Supports In-Line Assembly Code in C Source
- Includes IEEE-754 Compatible, High-Speed, Accelerated Floating-Point library for Components without On-Chip Floating-Point Instructions
- Supports Windows 95, DOS and Selected UNIX Hosts
- Source Code Supplied
- Annual Support Contracts Available

ASSEMBLER AND LINKER

The assembler processes assembly code produced by the compiler. The CTOOLS toolset offers other valuable utilities such as:

- Debugging aids: object file dumper and mapper
- An archiver to build libraries
- An object file stripper to eliminate debug records from the object module
- A COFF to IEEE-695 object file converter
- A big-endian to little-endian object file converter
- A ROM builder to produce ROMable code

The linker links together separately compiled modules, performing additional optimizations such as replacing calls by branch-and-link sequences. It reads the contents of a configuration file in order to map the application's code and data sections in memory and then link correct run-time libraries for the application. Linkage may be performed in interactive steps until the final link step, at which time all unresolved externals are satisfied.

DEBUGGER AND MONITOR

The gdb960 symbolic debugger and MON960 monitor are included in CTOOLS. The debugger is a full symbolic debugger, and operates with the MON960 monitor to allow setting of breakpoints, single-stepping, variable tracing, and many other capabilities.

LIBRARY SUPPORT

CTOOLS supports three library types:

- i960 architecture-specific high-level libraries
- IEEE-754 compatible accelerated floating-point libraries
- Low-level libraries supporting i960 processor evaluation boards

The CTOOLS linker configuration files hide the complexity of linking the correct libraries. All C libraries have been optimized and generated using the CTOOLS compiler. They are offered in normal code form, in position-independent form for use in applications relocatable at load time, and in big-endian form for applications that use i960 big-endian memory regions.

TECHNICAL SUPPORT

Annual software maintenance contracts are available from Intel. Contracts include free production upgrades, 1-800 technical support, FaxBack, BBS and guaranteed bug turnaround (once they have been identified). Intel also offers a 30-day, money-back guarantee to customers who are not satisfied after purchasing any Intel development tool.

HOST SYSTEMS SUPPORTED:

PC Compatible/DOS, Windows 95, HP9000/HP-UX, IBM RS6000/AIX, Sun-4

PROCESSORS SUPPORTED:

i960® Sx, Kx, Cx, Jx, RP, and Hx Processors

AVAILABILITY:

Now

CONTACT:

Local Distributor, Intel Sales Office or Intel Support at (800) 628-8686.

World Wide: call + 1 (503) 264-7354, 7-5:00, Mon-Wed & Fri; 7-3:00, Thur. All U.S. Pacific time.

WWW: http://www.intel.com

ORDERING INFORMATION:

CTOOLSW95KT	Windows 95 & PC Compatible/DOS — CD-ROM & 3 1/2" diskettes
CTOOLSDOSKT	PC Compatible/DOS — 3 1/2" diskettes & CD-ROM
CTOOLSUNXKT	HP9000/HP-UX — 4mm Sun-4/UNIX & IBM RS/6000/AIX — 8mm & QIC-24

i960® Microprocessor Literature

Title	Literature Order #	FaxBack Document #
PRODUCT INFORMATION		
i960® Processor Product Line Card		2033
i960® Processor Literature List		2115
FaxBack Document List		2068
i960® CA/CF 32-Bit Superscalar Microprocessor InfoGuide		2705
i960® KA/KB 32-Bit Embedded Microprocessor InfoGuide		2716
i960® HA/HD/HT Superscalar Microprocessor InfoGuide		2730
i960® JX Microprocessor/ The Cobra Series InfoGuide		2731
i960® SA/SB 32-Bit Embedded Microprocessors with 16-Bit Burst Data Bus	272233	
Enhanced PC I/O Performance with i960® RP Processor	272740	
DATA SHEETS		
80960HA/HD/HT 32-Bit High-Performance Superscalar Processor	272495	
80960JA/JF Embedded 32-Bit Microprocessor	272504	
80960CA-33,-25,-16 32-Bit High Performance Superscalar Processor	270727	
80960CF-33,-25,-16 32-Bit High Performance Superscalar Processor	272187	
80960KA Embedded 32-Bit Microprocessor	270775	
80960KB Embedded 32-Bit Microprocessor with Integrated Floating-Point Unit	270565	
80960SA Embedded 32-Bit Microprocessor with 16-Bit Burst Data Bus	272206	
80960SB Embedded 32-Bit Microprocessor with 16-Bit Burst Data Bus	272207	
82961KD Printer Coprocessor	272221	
80960 Intelligent I/O Microprocessor	272737	
APPLICATION NOTES/APPLICATION BRIEFS		
i960® Microprocessor Competitive Benchmark Report	272392	2515
Internetworking and the Intel i960® Microprocessor	272601	2359
Imaging and the Intel i960® Microprocessor	272602	2360
AB-42 80960Kx Self-Test	270703	
AP-506 Designing for 80960Cx and 80960Hx Compatibility	272556	
AP-703 DRAM Controller for the 33-MHz i960® CA/CF Microprocessor	272627	

Title	Literature Order #	FaxBack Document #
APPLICATION NOTES/APPLICATION BRIEFS (cont'd)		
AP-704 A Simple DRAM Controller for the i960® Cx Processor Using Flexlogic	272628	
AP-706 DRAM Controller for the 40-MHz i960® CA/CF Microprocessor	272655	
AP-712 DRAM Controller for the 33-MHz i960® JA/JF/JD Microprocessor	272674	
i960® RP Processor: A Single Chip Intelligent I/O Subsystem	272238	
TOOLS		
Technical Assistance (tools)		2544
Solutions960® Catalog	270791	
GNU/960 Software Toolset Fact Sheet	272178	
i960® Microprocessor Product Line and Support Tools Fact Sheet	272219	
EP80960CX Evaluation Platform	272505	
i960® Microprocessor Evaluation Platform/Cyclone EP	272508	
Cyclone* Evaluation Platform User's Guide	272577	
i960® SA/SB Processor Evaluation Board Fact Sheet	272033	
QT 960 Evaluation and Prototyping Board Fact Sheet	270743	
EV80960SX Evaluation Board User's Manual	270853	
EP80960CX Evaluation Platform User's Guide	272456	
82596CA High-Performance 32-Bit Local Area Network Coprocessor	290218	
MANUALS/DATABOOKS		
i960® Jx Microprocessor User's Manual	272483	
i960® Cx Microprocessor User's Manual	270710	
i960® KB Microprocessor Programmer's Reference Manual	270567	
i960® SA/SB Microprocessor Reference Manual	270929	
82961KD Printer Coprocessor Reference Manual	272280	
i960® Extended Architecture Programmer's Reference Manual	271191	
i960® Processors and Related Products Databook	272084	
i960® RP Microprocessor User's Manual	272736	

Intel Reference Numbers	
World Wide Web Address:	**http://www.intel.com**
FaxBack System:	**1 (800) 525-3019 or (503) 264-6835**
Application Bulletin Board System:	**1 (503) 264-7999**
Intel Literature Center:	**1 (800) 548-4725 7 a.m. to 7 p.m. CST**
Retail PC and Network Products:	**1 (800) 538-3373 or (503) 629-7000 7 a.m. to 7 p.m. PST**
General Information Hotline:	**1 (800) 628-8686 & (916) 356-3104 5 a.m. to 5 p.m. PST**

Intel Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in an Intel product. No other circuit patent licenses are implied. Information contained herein supercedes previously published specifications on these devices from Intel. *Other brands and names are the property of their respective owners. ©Intel Corporation 1996

UNITED STATES
Intel Corporation
2200 Mission College Blvd.
Santa Clara, CA
95052-8119

JAPAN
Intel Japan, K.K.
5-6 Tokodai,
Tsukuba-shi
Ibaraki-ken,
300-26

FRANCE
Intel Corporation
S.A.R.L.
1, Quai De Grenelle,
BP543
75725 Paris Cedex

UNITED KINGDOM
Intel Corporation (U.K.) Ltd.
Pipers Way
Swindon SN3 1RJ
Wiltshire,
England

GERMANY
Intel GmbH
Dornacher Strasse 1
D-85622 Felkirchen
Muenchen

HONG KONG
Intel Semiconductor Ltd.
32/F Two Pacific Place
88 Queensway,
Central

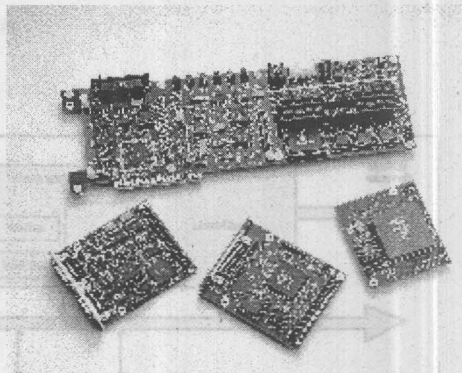
CANADA
Intel Semiconductor
of Canada, Ltd.
190 Attwell Drive,
Suite 500
Rexdale, Ontario M9W 6H8

Printed on Recycled Paper

Order Number 281434-003 Printed in U.S.A./0596/3K/IL KG

i960® Microprocessor Evaluation Platform

Cyclone EP



By tailoring these low cost modules to be interchangeable, designers are freed from the burdensome and time consuming task of supporting multiple boards during an architectural evaluation.

Flexibility Plus

Intel's Cyclone Evaluation Platform is designed to provide the developer with a flexible, low cost environment for code execution and software debug. Configured as a stand-alone base board with interchangeable modules, the Cyclone EP is capable of supporting the entire i960® processor family; from the low cost i960 SA processor to the newest and highest performance member of the i960 family, the i960 HT processor.

The standard interface incorporated on the Cyclone EP allows a designer to interchange application specific modules ranging from Ethernet to SCSI-3. Published specifications for the interface allow designers to build their own module or have Cyclone Microsystems custom develop it for them.

Processor Modules

Designed for the Cyclone EP are CPU modules featuring a specific i960 processor. By tailoring these four low cost modules to be interchangeable, the designer is freed from the burdensome and time consuming task of supporting multiple boards during an architectural evaluation. The CPU modules each contain a boot flash ROM with the MON960 monitor, appropriate glue logic and configuration switches. Using these switches the designer has the ability to further tune the design.

DRAM Memory

The Cyclone EP is shipped with 2 Mbytes of interleaved DRAM memory upgradable to 8 or 32 Mbytes. The DRAM controller automatically adjusts the wait-states based on processor type, clock frequency, and memory speed. The controller supports burst transfers using the interleaved banks to maximize performance. The default memory configuration comes standard with 70ns memory. For processor frequencies of 25 and 40 MHz, performance can be significantly increased by using 60ns memory. The board requires no shunts or switches to adjust for memory changes. The Presence Detect Signals and the controller automatically adjusts to minimum wait-states for the processor frequency and memory speeds. In addition the initialization code automatically sizes the memory so user software can take advantage of larger memory modules.

Software Development Support

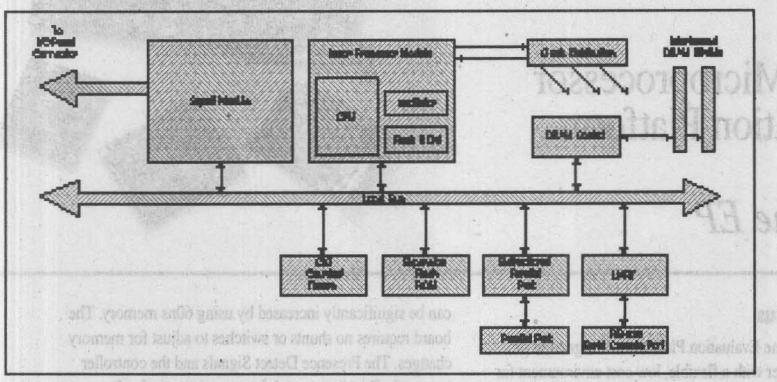
The Cyclone EP supports many software development tools. The installation instructions presented in the manual were verified using Intel's GNU/960 and IC960. These advanced C-language compilers for the i960 processor family are available for DOS-based systems and a variety of UNIX workstation hosts. Both products provide execution profiling and instruction scheduling optimizations for tuned code generation. In addition to the Intel tools, the Cyclone EP has been validated for use with many of the Solutions960® program development tools.

Console Serial Port/Parallel Port

The Cyclone EP has a single console port with an RS-232 line interface as well as a Centronix PC compatible input parallel port. Cables are supplied for both communication ports.

Cyclone Evaluation Platform Features

- Interchangeable i960 processor modules
- DIP switch selectable processor clock frequency
- 2 Mbytes of DRAM, expandable to 32 Mbytes
- DRAM controller automatically optimizes wait- states to processor frequency and memory
- Flash ROM sockets
- Parallel download port
- Three sixteen bit counter/timers
- Squall II module I/O expansion interface



Cyclone Evaluation Platform Functional Block Diagram

Squall II Module Interface

The Cyclone EP has a single expansion location (Squall II) and connector. While Cyclone Microsystems has many off the shelf modules available, users are also encouraged to build their own modules or contact Cyclone regarding custom modules and boards. Boards currently available through both Intel and Cyclone Microsystems are:

- Ethernet
- SCSI-2
- SCSI-3
- High Speed Serial

The Cyclone EP Functional Overview

I/O subsystem provides data buffers and simplified-control:

- The Centronics compatible parallel port allows fast download of code or data to the Cyclone EP The asynchronous serial (RS-232) port provides transfers up to 115.2 Kbaud.
- A Z8536 timer/counter provides three 16-bit counters, with interrupts.

A single expansion bus (Squall II module) allows expansion cards and external devices direct access to the i960 processor's bus and control signals.

Product Ordering Information

EP80960BB	Base Board without CPU Module
CPU80960HX[1]	HX Microprocessor CPU Module
CPU80960CX	CX Microprocessor CPU Module
CPU80960SX	SX Microprocessor CPU Module
CPU80960KX	KX Microprocessor CPU Module
CPU80960JX	JX Microprocessor CPU Module

[1] Available Q2 '95

I/O MODULES

SQETHERNET	ETHERNET MODULE
SQHSS[1]	HIGH SPEED SERIAL
SQSCSI-2[2]	SCSI-2
SQSCSI-3[2]	SCSI-3 (FAST AND WIDE)

[1] Not available on the CPU80960KX OR CPU80960SX

[2] Not available on the CPU80960SX

Support Information

Product and Sales Information	Intel product information, sales office and distributor phone numbers	**800-628-8686**
Literature Department	To order Intel product literature.	**800-548-4725**
Development Tools Hotline	Technical applications support for Intel development tools.	**800-843-4481**
FaxBack Service	Automated response system that faxes Intel documents to your fax machine at no cost. For a listing of all i960 microprocessor documents, order FaxBack service document 2068.	**800-628-2283**
Applications Bulletin Board Service	24-hour access via modem to publicly available technical information. For a complete listing of all files available on BBS, call the FaxBack service and request catalog 6 (BBS Master File Listing).	**916-356-3600**

Phone numbers listed above are for U.S. and Canada only.

UNITED STATES
Intel Corporation
2200 Mission College Boulevard
P.O. Box 58119
Santa Clara, CA 95052-8119

JAPAN
Intel Japan K.K.
5-6 Tokodai, Tsukuba-shi
Ibaraki, 300-26

FRANCE
Intel Corporation S.A.R.L.
1, Rue Edison, BP 303
78054 Saint-Quentin-en-Yvelines Cedex

UNITED KINGDOM
Intel Corporation (U.K.) Ltd.
Pipers Way
Swindon
Wiltshire, England SN3 1RJ

GERMANY
Intel GmbH
Dornacher Strasse 1
8016 Feldkirchen bei Muenchen

HONG KONG
Intel Semiconductor Ltd.
32/F Two Pacific Place
88 Queensway
Central

CANADA
Intel Semiconductor of Canada, Ltd.
190 Attwell Drive, Suite 500
Rexdale, Ontario M9W 6H8

Cyclone Microsystems, Inc.
25 Science Park
New Haven, CT
Phone: 203-786-5536
FAX: 203-786-5025
email: info@cyclone.com

Intel Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in an Intel product. No other circuit patent licenses are implied. Information contained herein supersedes previously published specifications on these devices from Intel.

*Other brands and names are the property of their respective owners

© Intel Corporation 1994

Order Number: 272598-001
Printed in the U.S.A./0994/15K/IL KG